Can LLM Rerankers Predict Their Own Ranking Performance?

Anonymous ACL submission

Abstract

Retrieval effectiveness varies substantially across queries, making it important to estimate ranking quality before relevance judgments are available. Query performance prediction (QPP) addresses this need, but most existing methods rely on external predictors after retrieval or reranking. In this paper, we study *reranker-internal QPP*: can an LLM reranker estimate the quality of the ranking it has just produced? We investigate both training-free and training-based approaches. For training-free estimation, we examine metric-specific self-consistency across sampled rankings and verbalized confidence produced directly by the reranker. Experiments on TREC Deep Learning 2019–2022 with four LLMs show that self-consistency is competitive with the state-of-the-art (SOTA) approach and better calibrated in almost all settings, while direct verbalized confidence is severely overconfident. To improve verbalized confidence, we propose two supervised methods, Verb-Num and Verb-List, which enable LLM rerankers to produce calibrated ranking-quality estimates with only a few additional output tokens.

1 Introduction

Retrieval effectiveness varies substantially across queries. Some queries are specific and well covered by the collection, while others are ambiguous, underspecified, affected by vocabulary mismatch, or have few relevant documents. A system that performs well on average can therefore still return poor results for particular queries, and these failures are especially consequential when top-ranked passages are shown to users or passed to retrieval-augmented LLMs. Query performance prediction (QPP) addresses this problem by estimating, before or without relevance judgments, how well a retrieval system is likely to perform for a given query (Carmel and Yom-Tov, 2010).

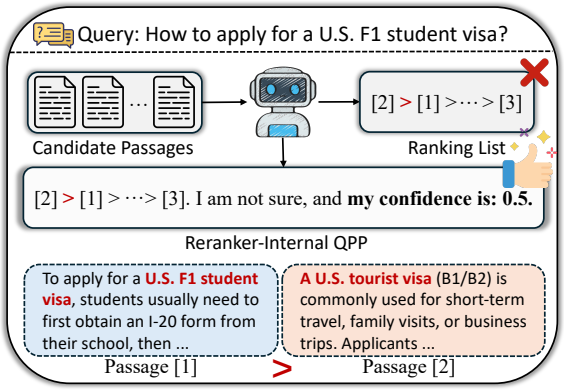


Figure 1: Overview of reranker-internal QPP. In this example, the model incorrectly ranks the less relevant Passage 2 ahead of Passage 1. We expect the reranker not only generates the ranked list but also expresses its confidence, helping prevent users from being misled by an erroneous ranking result.

Such estimates make retrieval more reliable and adaptive: they can identify difficult queries, trigger query reformulation or stronger reranking only when needed, allocate computation to uncertain cases, and support monitoring when manual relevance judgments are unavailable.

Existing QPP methods are commonly grouped into pre-retrieval and post-retrieval approaches (Carmel and Yom-Tov, 2010), with the latter usually stronger because they exploit the returned list. Unsupervised methods rely on signals such as query clarity (Cronen-Townsend et al., 2002), score distributions (Shtok et al., 2012; Pérez-Iglesias and Araujo, 2010; Tao and Wu, 2014), robustness, document coherence (Aslam and Pavlu, 2007; Arabzadeh et al., 2021a), and, for neural rankers, score uncertainty or representation-based agreement (Zendel et al., 2023; Vlachou and Macdonald, 2023; Singh et al., 2023). Supervised methods learn from labeled queries using neural or PLM-based representations of queries and ranked documents (Hashemi

et al., 2019; Arabzadeh et al., 2021b; Datta et al., 2022c). QPP-Gen (Meng et al., 2025) recently introduced LLM-based relevance judgments for QPP. Most of these methods, however, remain separate modules placed after retrieval or reranking.

This paper studies a different question: can an LLM reranker estimate the quality of the ranking it has just produced? We refer to this setting as **reranker-internal QPP**, emphasizing that the confidence signal comes from the reranker itself rather than from an external QPP model. The question is natural for sequence-to-sequence LLM rerankers, which take a query and candidate passages as input and directly generate a ranked list (Figure 1). If such a model can judge document relevance well enough to rank, it may also contain useful signals about when its ranking is likely to be reliable.

Reranker-internal QPP is closely related to confidence calibration in question answering (QA), where a calibrated model should assign confidence that matches answer correctness. Prior work has studied probability-based confidence (Guo et al., 2017; Jiang et al., 2021; Kadavath et al., 2022; Si et al., 2022), self-consistency (Manakul et al., 2023; Kuhn et al., 2023), and verbalized confidence (Tian et al., 2023; Xiong et al., 2023; Ni et al., 2024), often finding that LLMs are overconfident. QPP shares the same calibration goal, but ranking differs from QA: it evaluates an ordered list rather than a single answer, and quality is position-sensitive and metric-dependent, e.g., Precision, MRR, or NDCG. Thus, answer-level agreement does not directly transfer to rankings, and verbalized confidence must estimate performance over multiple ordered documents.

We investigate both training-free and training-based approaches. For training-free estimation, we study *self-consistency* across sampled rankings with metric-specific measures for Precision, MRR, and NDCG, and *verbalized confidence*, where the model directly outputs a numerical confidence score. We then study whether supervised training can teach an LLM reranker to express calibrated confidence while preserving efficient reranking.

Research Questions. We organize the study around three questions:

- **RQ1**: How predictive and calibrated are metric-specific self-consistency signals for LLM reranking performance?
- **RQ2**: Can LLM rerankers verbalize accurate confidence in their own ranking quality without training, and what failure modes arise?
- **RQ3**: Can supervised training enable LLM rerankers to output calibrated ranking-quality estimates?

Main Findings. Experiments on TREC DL 2019–2022 with LLaMA3.1 and Qwen2.5 models show that self-consistency is a strong training-free signal: it achieves competitive Spearman correlation with QPP-Gen and is better calibrated in almost all settings, though it can underperform on precision-oriented metrics. In contrast, direct verbalized confidence is poorly calibrated, with LLM rerankers assigning high confidence to most queries. We further propose two supervised methods: **Verb-Num**, which outputs a scalar estimate of a target ranking metric, and **Verb-List**, which outputs relevance indicators for top-ranked documents. Both improve QPP performance; Verb-Num tends to obtain stronger discriminability, while Verb-List is better calibrated and requires only a few additional output tokens.

Contributions. Our contributions are: 1) We identify the connection between QPP and confidence calibration in QA, and formulate reranker-internal QPP for sequence-to-sequence LLM rerankers. 2) We analyze two training-free confidence signals, metric-specific self-consistency and verbalized confidence, showing that the former is competitive and better calibrated than QPP-Gen, while the latter suffers from severe overconfidence. 3) We propose Verb-Num and Verb-List, two training methods that enable LLM rerankers to produce calibrated ranking-quality estimates with minimal additional generation cost.

2 Related Work

Query Performance Prediction Post-retrieval QPP estimates retrieval effectiveness without relevance labels. Existing methods are commonly divided into unsupervised and supervised approaches. Unsupervised methods exploit properties of retrieved results, such as clarity (Cronen-Townsend et al., 2002), robustness (Aslam and Pavlu, 2007), coherence (Arabzadeh et al., 2021a; Vlachou and Macdonald, 2023), ranking scores (Shtok et al., 2012), entropy (Zendel et al., 2023), and pairwise preferences (Singh et al., 2023). Supervised methods learn performance estimators from ranked lists, ranging

from neural models (Zamani et al., 2018; Datta et al., 2022a) to fine-tuned PLMs and multi-task frameworks (Arabzadeh et al., 2021b; Chen et al., 2022; Khodabakhsh and Bagheri, 2023). Recently, QPP-Gen (Meng et al., 2025) leverages LLMs to judge document relevance and achieves strong performance. Unlike prior methods that treat ranking and QPP as separate modules, we study whether an LLM reranker can perceive its own ranking performance.

LLM Knowledge Boundary Perception A reliable model should recognize when its outputs are likely to be wrong. This ability is often studied through confidence estimation, including probabilistic confidence from token likelihoods (Guo et al., 2017; Kadavath et al., 2022), self-consistency across generations (Manakul et al., 2023; Kuhn et al., 2023), verbalized confidence (Xiong et al., 2023; Tian et al., 2023), and confidence signals from internal representations (Azaria and Mitchell, 2023; Chen et al., 2024; Ni et al., 2025b). Our work extends this perspective to LLM reranking, asking whether the reranker can assess the quality of the ranking it just produced. Due to space limitations, we provide a more detailed discussion in § A.

3 Preliminary

Sequence-to-sequence Reranking. Following (Liu et al., 2025), for a query q, we first retrieve the top 100 passages $\mathcal{D} = \{d_1, d_2, \ldots, d_{100}\}$ using BM25 (Robertson et al., 1995). An LLM reranker then takes all passages as input and outputs a relevance-based ordering in text, as shown in Figure 10. We represent the resulting ranking as a sequence of indices $\mathcal{I} = \{i_1, i_2, \ldots, i_{100}\}$. For example, $i_1 = 5$ indicates that the 5th passage in the original input is ranked first after re-ranking.

QPP-Gen. Query performance prediction aims to assess the quality of a ranked list without ground-truth relevance annotations. QPP-Gen achieves SOTA performance among existing unsupervised QPP methods. It leverages an LLM to perform relevance judgments on each passage in the ranking list and calculates the corresponding ranking performance based on these judgments, as showing in Figure 2.

4 Experimental Setup

Ranking Quality Evaluation. We evaluate ranking quality with standard top-k IR metrics, including **Precision@k**, **MRR@k** (Mean Reciprocal Rank), and **NDCG@k**. Since sequence-to-sequence reranking only reorders the retrieved candidate pool $\mathcal{D}$, we compute NDCG with respect to the relevant passages within this pool and denote it as **NDCG-I** where "I" denotes "Input". Let $r_j \in \{0, 1\}$ be the binary relevance label of passage d_j, and let $\mathcal{R} = \{j : r_j = 1\}$ be the set of relevant passage indices in $\mathcal{D}$. Given the generated ranking $\mathcal{I} = \{i_1, \ldots, i_{100}\}$, the metrics are:

$$\text{NDCG-I@}k = \frac{\text{DCG@}k}{\text{IDCG-I@}k},$$

$$\text{DCG@}k = \sum_{m=1}^{k} \frac{\mathbb{I}(i_m \in \mathcal{R})}{\log_2(m+1)},$$

$$\text{IDCG-I@}k = \sum_{m=1}^{k} \frac{1}{\log_2(m+1)}.$$

Here, i_m denotes the passage index ranked at position m, and $\mathbb{I}(\cdot)$ is the indicator function.

QPP Evaluation Unlike prior QPP studies (Meng et al., 2025; Arabzadeh et al., 2021b), which primarily focus on discriminability, we argue that a good QPP method should also be well calibrated, i.e., its predicted scores should numerically reflect the actual query performance. Therefore, in addition to evaluating discriminability with the **Spearman correlation coefficient** (ρ) (Spearman, 1961), we introduce the **Expected Calibration Error (ECE)** (Guo et al., 2017) to explicitly assess calibration quality. ECE measures the discrepancy between predicted scores and true performance values. Specifically, samples are partitioned into B bins according to their predicted scores, and the calibration error is computed as the weighted average absolute difference between the average predicted score and the average true performance within each bin (B=10 in the paper):

$$\text{ECE} = \sum_{b=1}^{B} \frac{|S_b|}{S} \left|\text{real}(b) - \text{pred}(b)\right|, \quad (1)$$

where S_b denotes samples in bin b, and real(b) and pred(b) denote the average true and predicted scores in that bin, respectively. Lower ECE indicates better calibration.

Datasets. Following the previous study (Meng et al., 2025), we conduct experiments on TREC Deep Learning (TREC-DL) tracks from 2019 to 2022 (Craswell et al., 2025), which are standard benchmarks for passage ranking. We also binarize

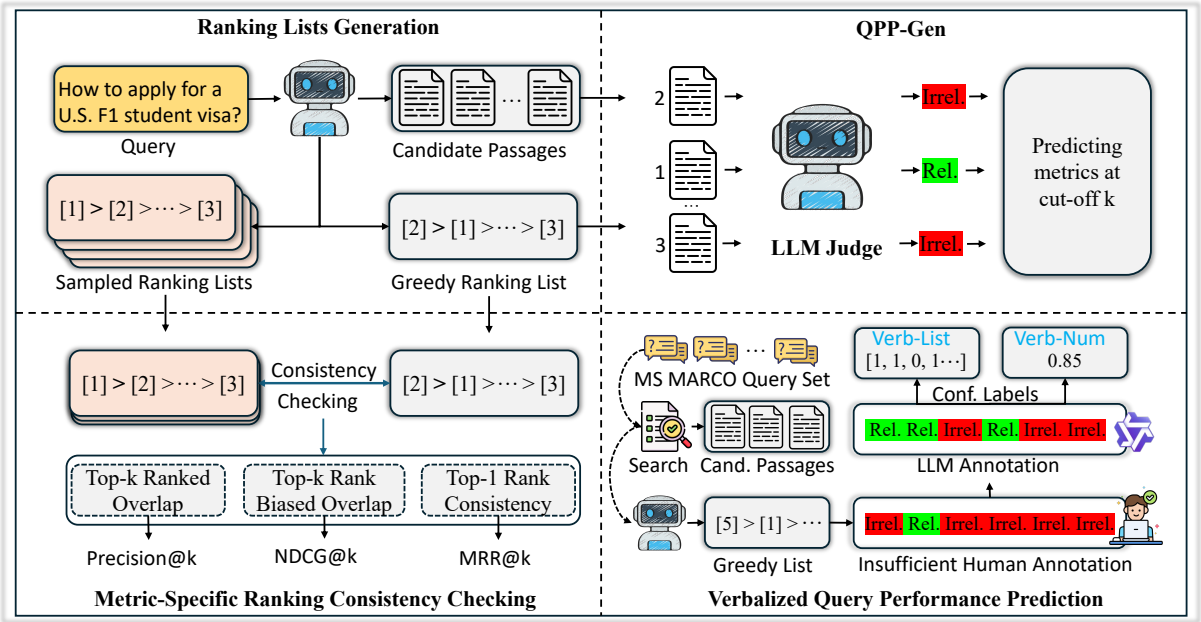


Figure 2: Overview of ranking lists generation, QPP-Gen, metric-specific consistency checking, and verbalized ranking-quality estimates calibration. QPP-Gen uses an LLM for pointwise relevance judgments. Self-consistency-based confidence measures agreement between multiple ranking lists under different IR metrics. Verbalized training labels are obtained by assessing rankings with both human annotation and LLM annotation.

relevance labels by treating passages with relevance ≥ 2 as positive. Details can be seen in § B.

Models. We use two representative series of LLMs, namely LLaMA3.1 (Grattafiori et al., 2024) and Qwen2.5 (Team, 2024). For LLaMA3.1, we use LLaMA3.1-8B-Instruct. To study the effect of model size, we leverage Qwen2.5-Instruct ranging from 7B to 32B.

5 QPP via Ranking Consistency

In this section, we answer how predictive and calibrated are metric-specific self-consistency signals for LLM reranking performance (**RQ1**).

5.1 Consistency between Two Ranking Lists

Measuring self-consistency between two ranking lists is non-trivial. First, ranking quality can be evaluated from multiple perspectives, a single consistency measure can not fully capture these distinct aspects. Second, it needs to account for positional differences. To address these challenges, we leverage the characteristics of established IR metrics to design metric-driven consistency measures.

Ranking Lists Generation. The model is asked to generate two types of ranking lists:

- **Greedy Ranking** ($\mathcal{I}_g$): Generated using greedy decoding (temperature $T = 0$). This is used for ranking performance evaluation.
- **Sampled Rankings** ($\mathcal{I}_{s,j}$): Generating $N = 20$ ranking lists by setting $T = 1$, where s denotes "sampling" and $j \in \{1, \ldots, N\}$.

Metric-Specific Ranking Consistency Checking. For each IR metric, we design a specific consistency function $\text{Consis}(\mathcal{I}_1, \mathcal{I}_2)$ to quantify the similarity between two ranking lists $\mathcal{I}_1$ and $\mathcal{I}_2$.

- **Consistency for Precision@k**: Since Precision considers the top-k results as an unordered set, we measure its consistency by the overlap ratio of the top-k items:

$$\text{Consis}_{\text{Prec}}(\mathcal{I}_1, \mathcal{I}_2) = \frac{|\mathcal{I}_1^{1:k} \cap \mathcal{I}_2^{1:k}|}{k}.$$

- **Consistency for NDCG-I@k**: To align with the position-based discounting, we adopt Rank Biased Overlap (RBO) (Webber et al., 2010). RBO evaluates the similarity of two lists by calculating the *agreement* A_d at each depth d, which represents the proportion of shared items within the top-d items:

$$A_d = \frac{|\mathcal{I}_{1,1:d} \cap \mathcal{I}_{2,1:d}|}{d},$$

where $\mathcal{I}_{1,1:d}$ and $\mathcal{I}_{2,1:d}$ are the sets of the first d documents in each list. As we evaluate ranking performance at a fixed depth k, we employ

a truncated version of RBO and normalize the score by the total weight of the first k terms:

$$\text{Consis}_{\text{NDCG-I@k}}(\mathcal{I}_1, \mathcal{I}_2) = \frac{1-p}{1-p^k} \sum_{d=1}^{k} p^{d-1} \cdot A_d,$$

where $p \in (0, 1)$ is the parameter (i.e., $p = 0.9$ in this paper) that determines the weight concentration on higher-ranked items; a smaller p assigns more weight to items at earlier positions and $1 - p^k$ is used for normalization.

- **Consistency for MRR@k**: Mirroring MRR's sensitivity to the position of the first relevant document, we view the first-ranked item in ($\mathcal{I}_1$) as relevant and define consistency as the reciprocal rank of the top-ranked item of the reference list ($\mathcal{I}_1$) within the compared list ($\mathcal{I}_2$):

$$\text{Consis}_{\text{MRR@k}}(\mathcal{I}_1, \mathcal{I}_2) = \frac{1}{\text{Rank}(i_{1,1}, \mathcal{I}_2)} \quad (2)$$

where $i_{1,1}$ is the item ranked first in $\mathcal{I}_1$, and $\text{Rank}(i_{1,1}, \mathcal{I}_2)$ denotes its rank index in $\mathcal{I}_2$.

Final Consistency Score. Consistency on a query is computed by the average consistency between the stable greedy ranking $\mathcal{I}_g$ and each of the N sampled rankings $\mathcal{I}_{s,j}$. We expect that a model with a clear understanding of ranking should achieve both high self-consistency and strong ranking performance.

5.2 Baselines

We compare self-consistency with QPP-Gen (Meng et al., 2025), a SOTA unsupervised QPP baseline, illustrated in Section 3. **Why not include traditional unsupervised QPP methods?** We focus on QPP-Gen as a strong and representative LLM-based method. Traditional score- or embedding-based QPP methods are less applicable to sequence-to-sequence reranking, where LLMs output only ranked lists without calibrated scores or query–document embeddings. Robustness-based QPP could be adapted by perturbing queries and measuring ranking stability, but this would be a separate consistency-based variant and is left for future work.

5.3 Results and Analysis

The ranking performance of each model and the self-consistency scores are shown in Table 3. The performance of self-consistency and QPP-Gen is shown in Table 1. Results on each dataset can be seen in Table 5. We find:

1) Self-consistency is better calibrated than QPP-Gen. As shown in Table 1, compared with QPP-Gen, the ECE corresponding to self-consistency is lower in almost all cases. To further understand this phenomenon, we analyze the distributions of the confidence scores predicted by the two methods, as shown in Figure 3. It can be observed that the confidence predicted by both methods is generally higher than the actual ranking performance. In particular, QPP-Gen assigns high confidence to more samples, indicating that this method incorrectly judges many highly ranked but irrelevant documents as relevant, which explains its worse ECE.

2) Self-consistency achieves competitive correlation with ranking performance. As shown in Table 1, compared with QPP-Gen, Self-consis achieves competitive Spearman correlations in most cases. Notably, Self-consis performs better on NDCG-I@10, while it can underperform on Precision and MRR. This contrast stems from the different limitations of the two methods.

QPP-Gen is not well-suited for recall-aware metrics. Since top-ranked passages are more likely to be relevant, false-positive errors of QPP-Gen are less emphasized at higher ranks. Consequently, QPP-Gen is naturally well suited for precision-oriented metrics such as Precision@k and MRR. However, recall-aware metrics such as NDCG-I@10 also depend on judgments of lower-ranked passages, where false positives tend to accumulate and reduce predictive accuracy. As a result, QPP-Gen performs worse than self-consistency on NDCG-I@10.

Self-consistency has a structural mismatch with ranking quality in relevance-dense pools. When many relevant documents are available, the model may place different relevant subsets at the top across runs, yielding high ranking quality but low consistency. As a result, self-consistency can underperform QPP-Gen on precision-oriented metrics. Figure 4 supports this observation: as the number of relevant documents increases, Precision@k tends to improve, whereas self-consistency may decline. So stability is not a guarantee of correctness for each query. A model can repeatedly make the same relevance mistake while remaining highly consistent. Self-consistency also requires multiple ranking runs, making it less efficient than QPP-Gen.

Table 1: Average QPP performance over TREC DL19–22. Bolds denote the best scores.

Model	Method	NDCG-I@10		Precision@10		MRR@10	
		Spearman ↑	ECE ↓	Spearman ↑	ECE ↓	Spearman ↑	ECE ↓
Llama3.1-8B	QPP-Gen	0.268	0.389	**0.436**	0.496	0.180	0.282
	Self-Consis	**0.337**	**0.179**	0.379	**0.166**	**0.225**	**0.221**
Qwen2.5-7B	QPP-Gen	0.289	0.200	0.403	**0.154**	**0.260**	0.261
	Self-Consis	**0.336**	**0.171**	**0.424**	0.166	0.204	**0.219**
Qwen2.5-14B	QPP-Gen	0.371	0.208	**0.548**	0.249	–	0.215
	Self-Consis	**0.373**	**0.158**	0.423	**0.158**	**0.307**	**0.202**
Qwen2.5-32B	QPP-Gen	0.295	0.212	**0.544**	0.251	**0.260**	0.229
	Self-Consis	**0.394**	**0.150**	0.454	**0.171**	0.210	**0.197**

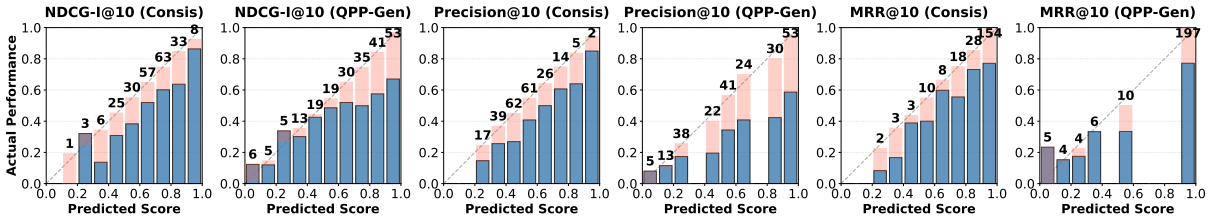
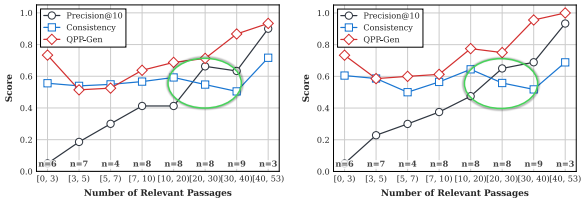


Figure 3: ECE of self-consistency and QPP-Gen on Qwen2.5-14B-Instruct. In each plot, the x-axis represents bins of predicted scores, while the y-axis shows the true performance of samples within each bin. The diagonal line $y = x$ indicates perfect calibration. The numbers above the bars denote the count of samples whose predicted confidence falls into each bin. Results are aggregated over DL19–22.



(a) Qwen2.5-14B (b) Qwen2.5-32B

Figure 4: Trends of self-consistency and QPP-Gen predicted scores for Qwen2.5-14B and 32B on DL21 with respect to the number of relevant passages in the input.

6 QPP via Verbalized Confidence

In this section, we answer **RQ2**: Can LLM rerankers verbalize accurate confidence in their own ranking quality without training, and what failure modes arise?

6.1 Experimental Setup

The model is asked to output its confidence that reflects a given metric such as Precision@10 while ranking. The output format is: `[2] > [1] > ··· > [5]. Confidence: 0.85.` We study Qwen2.5-series models. The other settings are the same as those in § 5.

6.2 Results and Analysis

Since the model is required to perform both ranking and confidence estimation, we first investigate whether this dual-task setup harms ranking performance. Results in Table 6 show that simultaneously performing both tasks does not significantly affect the model's ranking performance.

Answer to RQ2: LLMs can not express their ranking performance in words accurately and are severely overconfident. We analyze the relation between verbalized confidence predicted by models of varying sizes and actual performance for Precision@10 and NDCG-I@10. The results are presented in Figure 5. We observe that, LLMs tend to assign higher verbalized confidence across all metrics. For most samples, the verbalized confidence exceeds 0.8, indicating severe overconfidence, which aligns with previous findings on verbalized confidence in LLMs' knowledge boundary perception (Xiong et al., 2023; Ni et al., 2024). As a result, the model's confidence diverges significantly from actual performance, leading to a high ECE and poor Spearman correlation. Compared to self-consistency-based QPP (See Table 1), verbalized QPP is significantly worse. In addition, this phenomenon is not effectively mitigated as the model size increases.

7 Teaching LLMs to Express Their Ranking Performance

In this section, we aim to answer **RQ3**: Can supervised training enable LLM rerankers to output calibrated ranking-quality estimates?

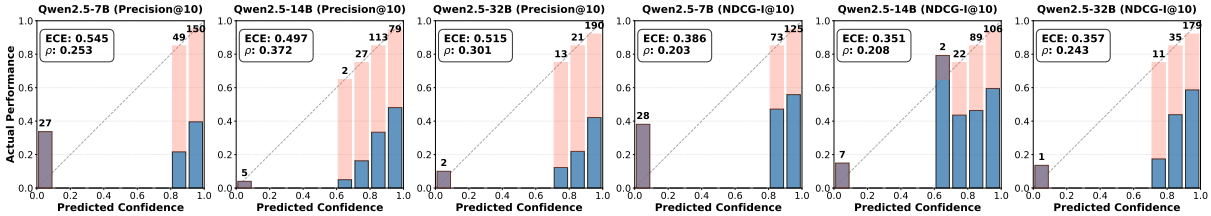


Figure 5: Comparison between verbalized confidence and true performance for the Qwen2.5 series models on the TREC DL19-22 datasets. ρ represents the Spearman correlation coefficient.

7.1 Method

Training Objective. To enable LLM reranker to perceive its own ranking quality, we construct a training set $\mathcal{T} = \{x_j, y_j\}_{j=1}^{N}$ where N is the count of training samples. For each instance, the input x_j consists of a query q_j and candidate passages D_j. The target y_j is a sequence formatted as: `<answer>` I_j `</answer>` `<confidence>` c_j `</confidence>`. Specifically, I_j is the model's self-generated ranking list for the input x_j and c_j represents the model's ranking performance, calculated by evaluating I_j against the ground-truth relevance labels.

We consider two forms of c_j: 1) **Verb-Num**: a single scalar value (e.g., "0.85") representing a specific IR metric. 2) **Verb-List**: an ordered list of binary indicators (e.g., "[1, 0, 1]") representing the ground-truth relevance of each passage in I_j, which is metric-agnostic. Label construction overview can be seen in Figure 2. The training loss is defined as:

$$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \left(\mathcal{L}_{\text{rank}}(x, y) + \lambda \mathcal{L}_{\text{conf}}(x, y)\right), \tag{3}$$

where λ is a balancing parameter used to optimize confidence expression without degrading the model's original ranking capability. Following (Liu et al., 2025), we employ an importance-aware loss $\mathcal{L}_{\text{rank}}$ to emphasize the learning of top-ranked passages. $\mathcal{L}_{\text{conf}}$ is the standard cross-entropy loss calculated over the tokens within the `<confidence>` tags. Details can be found in § C.

Insufficient Human Annotation Augmentation. We leverage the widely used MS MARCO training set (Bajaj et al., 2016) as our training data. Although large-scale, its ground-truth annotations are sparse, which can lead to inaccurate training signals. To address this issue, we use the powerful reasoning model Qwen3-32B (Yang et al., 2025) to augment the relevance annotations (See Figure 8). For efficiency, we annotate only the top-10 passages ranked by each model for the first 40k queries. Further details are provided in § C.1.

7.2 Experimental Setup

Due to space limitation, see details in § C.

Models. We evaluate confidence verbalization under two training settings. The first setting assumes access to passage-level relevance annotations, which allows us to improve both the reranker's ranking ability and its verbalized QPP ability. Following prior work (Liu et al., 2025), we first enhance Qwen2.5-7B-Instruct with GPT-4o-generated ranking lists, and then train it to express the quality of its own rankings. We denote the resulting ranking-enhanced model as Qwen2.5-7B-Instruct100. The second setting considers scenarios such as RAG, where passage-level relevance annotations may be unavailable and only an overall ranking-quality score can be obtained. In this case, we do not enhance the model's ranking ability, but only train it to verbalize its ranking quality. We refer to this original model as Qwen2.5-7B-Instruct.

Metrics. We use normalized **DCG@10** for ranking performance, and Spearman correlation and ECE for predicted confidence, where DCG@10 is normalized by the ideal score to keep confidence values within $[0, 1]$.

Datasets. We train on 20k balanced queries and evaluate on 2,500 held-out MS MARCO queries in-domain and TREC DL 2019–2022 OOD.

Baselines. We compare a range of SOTA unsupervised and supervised QPP methods. For unsupervised methods, we include **QPP-Gen** (Meng et al., 2025). For supervised methods, we consider: **BERTQPP** (Arabzadeh et al., 2021b): a regression-based method that predicts QPP scores from BERT representations of the query and top-ranked document. **qppBERT-QL** (Datta et al., 2022c): splits the ranked list into chunks, predicts the number of relevant items per chunk, and averages the results with weighted importance. **QPP-**

Table 2: QPP performance on Trec DL19-22 datasets. Bold and underline indicate the best and second-best results per ranking model. Statistical significance ($p < 0.05$) is denoted by * for improvement over QPP-Gen-SFT (supervised on MS MARCO) and † for improvement over QPP-Gen (3-32B). Significance for Spearman is calculated via Steiger's Z-test, and for ECE via a paired t-test.

Rankers	Methods	Spearman ↑					ECE ↓				
		DL19	DL20	DL21	DL22	Avg.	DL19	DL20	DL21	DL22	Avg.
Qwen2.5-7B-Instruc100	*Unsupervised*										
	QPP-Gen	0.178	0.479	0.347	0.497	0.397	0.288	**0.152**	0.234	0.176	0.205
	QPP-Gen (3-32B)	0.407	0.534	0.273	0.669	0.494	0.212	0.239	0.233	0.193	0.217
	Supervised on MS MARCO										
	BERTQPP	0.030	-0.022	0.103	0.004	0.026	0.564	0.444	0.524	0.300	0.437
	qppBERT-PL	0.164	0.233	0.189	0.105	0.167	0.587	0.460	0.588	0.341	0.474
	QPP-Gen-SFT	0.253	0.477	0.450	0.462	0.423	0.230	0.297	0.293	0.362	0.305
	Supervised on Augmented MS MARCO										
	BERTQPP	0.318	0.132	0.402	0.322	0.295	0.534	0.414	0.494	0.274	0.409
	qppBERT-PL	**0.499**	0.514	0.253	0.216	0.350	0.516	0.423	0.599	0.377	0.466
	QPP-Gen-SFT	0.258	0.434	0.223	0.399	0.339	0.296	0.401	0.342	0.508	0.403
	Verb-Num	0.385	**0.626**	0.475†	**0.679***	**0.563**	0.510	0.391	0.455	0.287	0.394
	Verb-List	0.423	0.605	**0.486**†	0.612	0.545	**0.166**	0.216*†	**0.215***	**0.163***	**0.188**

Gen-SFT (Meng et al., 2025): fine-tunes an LLM to improve pointwise relevance judgments.

7.3 Results and Analysis

Table 2 reports the main results under the passage-level annotation setting, where the model is first enhanced for ranking and then trained for verbalized QPP. Table 9 reports the results under the ranking-quality-only setting, where the model is trained to verbalize ranking quality without additional ranking enhancement. Our findings are as follows. **1) Verb-Num and Verb-List achieve SOTA QPP performance.** As shown in Table 2, the two methods we propose outperform existing approaches on most datasets, even surpassing the powerful Qwen3-32B. Specifically, Verb-Num often achieves the best Spearman correlation, while Verb-List attains the lowest ECE. This is consistent in in-domain evaluation (See § C.2). Compared with QPP-Gen, which requires separate inference over each retrieved passage, verbalized QPP introduces minimal overhead by generating only a few additional tokens after ranking. **2) Compared to Verb-Num, Verb-List is better calibrated but exhibits weaker correlation.** One possible reason is that Verb-List only needs to learn relevance judgments, without explicitly modeling complex metric computations. Consequently, its predicted confidence is numerically closer to the true performance. However, because the model treats relevance judgments at different positions equally, it fails to capture positional importance, which may lead to lower correlation. Incorporating positional weights is a promising direction for future work. Further calibration of Verb-Num could likely be addressed by using more training data.

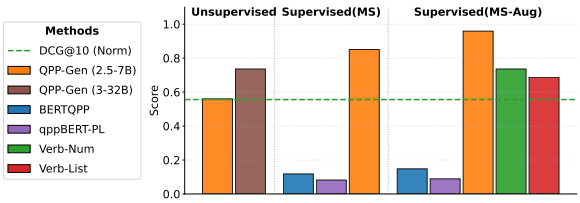


Figure 6: Average predicted scores of different methods on DL19–22 using ranking lists from Verb-Num (Qwen2.5-7B-Instruct100). The horizontal line indicates ranking performance. MS and MS-Aug denote training on the original and augmented MS MARCO annotations, respectively.

The ranking-quality-only setting shows a similar trend, suggesting that our method can still improve verbalized QPP when only aggregate ranking-quality supervision is available. Moreover, as shown in Figure 6, Qwen3-32B can produce false positives, leading to overestimate ranking performance. Details are in § C.2.

8 Conclusion

In this paper, we study reranker-internal QPP, asking whether LLM rerankers can estimate the quality of their own generated rankings. We find that LLMs' self-consistency across multiple ranking generations is competitive with the SOTA unsupervised method QPP-Gen, but they still struggle to express confidence in a human-like verbalized form. To address this, we propose two training methods, Verb-Num and Verb-List, to improve verbalized confidence. Future directions include: (1) modeling ranking performance with multi-level relevance, (2) improving reranker-internal QPP in an unsupervised setting, and (3) exploring whether long-horizon reasoning further benefit reranker-internal QPP.

Limitations

First, we focus on evaluating ranking performance with relatively simple binary relevance settings, leaving reranker-internal QPP under multi-level relevance unexplored. Second, although Verb-Num and Verb-List improve verbalized confidence through training, we do not fully address how to improve reranker-internal QPP in an unsupervised setting. Third, we do not examine whether long-horizon reasoning and reflection can further benefit LLM rerankers in perceiving their own ranking performance. These limitations point to important directions for future research.

Ethical Considerations

All models used in this study are open-source LLMs, and all datasets are derived from publicly available open-source resources. We use these models and datasets in accordance with their respective licenses and intended research purposes. Our experiments are conducted solely for academic research and do not involve human subjects, private user data, or sensitive personal information. Since our work focuses on query performance prediction and model self-awareness in ranking tasks, it does not introduce additional risks related to harmful content generation or real-world deployment. AI assistants were used solely for grammar or style polishing. No original research ideas, experimental designs, analyses, or scientific claims were generated by AI tools. All substantive intellectual contributions and final manuscript content were produced, reviewed, and verified by the authors.

References

Negar Arabzadeh, Amin Bigdeli, Morteza Zihayat, and Ebrahim Bagheri. 2021a. Query performance prediction through retrieval coherency. In *European Conference on Information Retrieval*, pages 193–200. Springer.

Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021b. Bert-qpp: contextualized pre-trained transformers for query performance prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2857–2861.

Javed A Aslam and Virgil Pavlu. 2007. Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *European conference on information retrieval*, pages 198–209. Springer.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. *arXiv preprint arXiv:2304.13734*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, and 1 others. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

David Carmel and Elad Yom-Tov. 2010. *Estimating the query difficulty for information retrieval*. Morgan & Claypool Publishers.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

Xiaoyang Chen, Ben He, and Le Sun. 2022. Group-wise query performance prediction with bert. In *European Conference on Information Retrieval*, pages 64–74. Springer.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M Voorhees, and Ian Soboroff. 2025. Overview of the trec 2022 deep learning track. *arXiv preprint arXiv:2507.10865*.

Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. 2002. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306.

Ronan Cummins, Joemon Jose, and Colm O'Riordan. 2011. Improved query performance prediction using standard deviation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1089–1090.

Suchana Datta, Debasis Ganguly, Derek Greene, and Mandar Mitra. 2022a. Deep-qpp: A pairwise interaction-based deep learning model for supervised query performance prediction. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 201–209.

Suchana Datta, Debasis Ganguly, Mandar Mitra, and Derek Greene. 2022b. A relative information gain-based query performance prediction framework with generated query variants. *ACM Transactions on Information Systems*, 41(2):1–31.

Suchana Datta, Sean MacAvaney, Debasis Ganguly, and Derek Greene. 2022c. A'pointwise-query, listwise-document'based query performance prediction approach. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2148–2153.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *arXiv preprint arXiv:2402.10612*.

Guglielmo Faggioli, Thibault Formal, Simon Lupart, Stefano Marchesin, Stephane Clinchant, Nicola Ferro, and Benjamin Piwowarski. 2023. Towards query performance prediction for neural information retrieval: challenges and opportunities. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 51–63.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2019. Performance prediction for non-factoid question answering. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 55–58.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Maryam Khodabakhsh and Ebrahim Bagheri. 2023. Learning to rank and predict: Multi-task learning for ad hoc retrieval and query performance prediction. *Information Sciences*, 639:119015.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Wenhan Liu, Xinyu Ma, Yutao Zhu, Ziliang Zhao, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2025. Sliding windows are not the end: Exploring full ranking with long-context large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 162–176.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of EMNLP 2023*, pages 9004–9017.

Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. 2025. Query performance prediction using relevance judgments generated by large language models. *ACM Transactions on Information Systems*, 43(4):1–35.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation. *arXiv preprint arXiv:2402.11457*.

Shiyu Ni, Keping Bi, Jiafeng Guo, Minghao Tang, Jingtong Wu, Zengxin Han, and Xueqi Cheng. 2025a. Annotation-efficient universal honesty alignment. *arXiv preprint arXiv:2510.17509*.

Shiyu Ni, Keping Bi, Jiafeng Guo, Lulu Yu, Baolong Bi, and Xueqi Cheng. 2025b. Towards fully exploiting llm internal states to enhance knowledge boundary perception. *arXiv preprint arXiv:2502.11677*.

Joaquín Pérez-Iglesias and Lourdes Araujo. 2010. Standard deviation as a query hardness estimator. In *International Symposium on String Processing and Information Retrieval*, pages 207–212. Springer.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, and 1 others. 1995. *Okapi at TREC-3*. British Library Research and Development Department.

Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems (TOIS)*, 30(2):1–35.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*.

Ashutosh Singh, Debasis Ganguly, Suchana Datta, and Craig McDonald. 2023. Unsupervised query performance prediction for neural models with pairwise rank preferences. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2486–2490.

Charles Spearman. 1961. The proof and measurement of association between two things.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. *arXiv preprint arXiv:2403.06448*.

Yongquan Tao and Shengli Wu. 2014. Query performance prediction by considering score magnitude and variance together. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1891–1894.

Qwen Team. 2024. Qwen2. 5: A party of foundation models.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Maria Vlachou and Craig Macdonald. 2023. On coherence-based predictors for dense query performance prediction. *arXiv preprint arXiv:2310.11405*.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023. Alignment for honesty. *arXiv preprint arXiv:2312.07000*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

Hamed Zamani, W Bruce Croft, and J Shane Culpepper. 2018. Neural query performance prediction using weak supervision from multiple signals. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 105–114.

Oleg Zendel, Binsheng Liu, J Shane Culpepper, and Falk Scholer. 2023. Entropy-based query performance prediction for neural information retrieval systems.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024. R-tuning: Instructing large language models to say 'i don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7106–7132.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley A Malin, and Sricharan Kumar. 2023. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. *arXiv preprint arXiv:2311.01740*.

Yun Zhou and W Bruce Croft. 2006. Ranking robustness: a novel framework to predict query performance. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 567–574.

Yun Zhou and W Bruce Croft. 2007. Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 543–550.

A Related Work

A.1 Query Performance Prediction

Query performance prediction can be broadly categorized into pre-retrieval and post-retrieval methods. In this paper, we focus on post-retrieval methods for their effectiveness. Existing post-retrieval QPP approaches can be broadly categorized into unsupervised and supervised methods.

Unsupervised Methods. A wide range of unsupervised QPP techniques have been proposed to estimate retrieval effectiveness, primarily for traditional lexical rankers such as BM25 (Robertson et al., 1995) and query likelihood (Lafferty and Zhai, 2001). These methods typically exploit statistical properties of the retrieved results and can be grouped into several categories: clarity-based methods (Cronen-Townsend et al., 2002), which measure the divergence between query and collection language models; robustness-based methods (Aslam and Pavlu, 2007; Zhou and Croft, 2006, 2007), which assess the stability of retrieval results under query perturbations; coherence-based methods (Arabzadeh et al., 2021a), which evaluate semantic consistency among top-ranked documents; and score-based methods (Shtok et al., 2012; Pérez-Iglesias and Araujo, 2010; Cummins et al., 2011; Tao and Wu, 2014), which derive predictive signals from the distribution of retrieval scores. However, post-retrieval QPP methods originally developed for lexical rankers have been shown to be less effective for neural ranking models. This limitation has motivated a line of unsupervised QPP methods specifically designed for neural rankers. Representative approaches include the weighted relative information gain (WRIG) model (Datta et al., 2022b), which estimates relative performance differences between a given query and its variants; entropy-based methods (Zendel et al., 2023), which exploit the score distributions produced by neural rankers; neural-specific regularization schemes (Faggioli et al., 2023); coherence-based methods leveraging dense representations (Vlachou and Macdonald, 2023); and pairwise preference-based methods (Singh et al., 2023), which quantify agreement across different neural rankers.

Supervised Methods. In parallel, supervised QPP methods learn mappings from ranked lists to performance estimates using labeled data (Zamani et al., 2018; Hashemi et al., 2019; Arabzadeh et al., 2021b; Datta et al., 2022a; Chen et al., 2022; Datta et al., 2022c; Khodabakhsh and Bagheri, 2023). Early neural approaches, such as NeuralQPP (Zamani et al., 2018) and Deep-QPP (Datta et al., 2022a), train predictive models from scratch, whereas later methods leverage pre-trained language models. For example, NQA-QPP (Hashemi et al., 2019) and BERT-QPP (Arabzadeh et al., 2021b) fine-tune BERT (Devlin et al., 2019). Subsequent work, including qppBERT-PL (Datta et al., 2022c), incorporates listwise document information, while BERT-groupwise-QPP (Chen et al., 2022) models both cross-query and cross-document dependencies. M-QPPF (Khodabakhsh and Bagheri, 2023) further adopts a multi-task learning framework that jointly optimizes document ranking and QPP. More recently, Meng et al. (Meng et al., 2025) proposed QPP-Gen, which uses an LLM to judge the relevance of documents in a ranked list with respect to a query. QPP-Gen demonstrates strong performance in unsupervised settings and can be further improved through supervised training, achieving state-of-the-art results.

Most existing QPP methods treat retrieval and QPP as two separate modules. Although M-QPPF jointly models these tasks, it remains constrained within the traditional BERT-based retrieval paradigm. In contrast, this paper aims to explore a unified paradigm we term **Self-aware QPP**, where the reranker serves as its own quality estimator.

A.2 LLM Knowledge Boundary Perception

The perception of knowledge boundaries in LLMs has been extensively studied in the question-answering (QA) domain. A reliable model should be able to recognize what it knows and what it does not, which is typically assessed by examining whether the model's confidence in its outputs aligns with its actual performance. Existing research on confidence estimation can be broadly categorized into four approaches.

Probabilistic Confidence (Guo et al., 2017; Desai and Durrett, 2020; Jiang et al., 2021; Kadavath et al., 2022; Si et al., 2022). Early work measured confidence through the model's generation probabilities. For classification tasks, Guo et al. (Guo et al., 2017) observed that ResNet (He et al., 2016) often assigns probabilities higher than the actual accuracy of its predictions and proposed temperature scaling to correct this. Subsequent studies (Desai and Durrett, 2020) found that pre-trained models such as BERT (Devlin et al., 2019) are generally well-calibrated. With the rise of generative LLMs, later research (Jiang et al., 2021; Kadavath et al., 2022; Si et al., 2022) revealed that models in free-form generation tasks tend to exhibit overconfidence.

Self-consistency-based Confidence (Fomicheva et al., 2020; Manakul et al., 2023; Kuhn et al., 2023; Zhang et al., 2023; Ding et al., 2024). Prob-

abilistic confidence can be influenced by generation format and is not applicable for black-box models. To address this, recent studies leverage the self-consistency of multiple model generations to estimate confidence: the more consistent the outputs across generations, the higher the model's confidence. Early approaches (Fomicheva et al., 2020) quantified consistency via term-level overlap between outputs, while recent work (Manakul et al., 2023; Kuhn et al., 2023) uses LLMs to assess semantic consistency, achieving state-of-the-art performance among unsupervised methods. Zhang et al. (Zhang et al., 2023) and Ding et al. (Ding et al., 2024) extended this approach by measuring consistency across different query variants and languages.

Verbalized Confidence (Yin et al., 2023; Tian et al., 2023; Xiong et al., 2023; Ni et al., 2024; Lin et al., 2022; Yang et al., 2023; Zhang et al., 2024). The capabilities of LLMs enable a new paradigm where models can express confidence in natural language. Researchers have studied whether LLMs can accurately verbalize confidence, examining both binary (Yin et al., 2023; Ni et al., 2024) and fine-grained confidence levels (Xiong et al., 2023; Tian et al., 2023), finding that models are highly overconfident. Parallel work has focused on training models to express confidence more accurately, demonstrating that LLMs can be taught to convey their certainty in words (Lin et al., 2022; Yang et al., 2023; Zhang et al., 2024).

Internal-state-based Confidence (Azaria and Mitchell, 2023; Su et al., 2024; Chen et al., 2024; Ni et al., 2025b,a). Beyond explicit verbalization, some studies show that a model's internal states inherently encode information about answer correctness. Lightweight networks can extract this information (Azaria and Mitchell, 2023; Su et al., 2024; Chen et al., 2024). Ni et al. (Ni et al., 2025b) found that hidden states prior to answer generation already encode confidence, and more recent work (Ni et al., 2025a) introduces a pre-training–finetuning paradigm for confidence estimation, enabling efficient development of estimators that generalize across domains.

In this paper, we explore whether LLMs can recognize their own ranking performance which is fundamentally different compared with LLMs' self-awareness in answer factuality.

Table 3: Average ranking performance and self-consistency results across TREC DL19-22 datasets. The averages are weighted by the number of queries in each dataset.

Metrics	Models	Ranking Scores	Consistency
NDCG_{100}@10	BM25	0.374	-
	Llama3.1-8B	0.493	0.642
	Qwen2.5-7B	0.496	0.641
	Qwen2.5-14B	0.515	0.664
	Qwen2.5-32B	**0.555**	**0.700**
Precision@10	BM25	0.293	-
	Llama3.1-8B	0.346	0.508
	Qwen2.5-7B	0.349	0.511
	Qwen2.5-14B	0.356	0.500
	Qwen2.5-32B	**0.385**	**0.537**
MRR	BM25	0.511	-
	Llama3.1-8B	0.703	0.877
	Qwen2.5-7B	0.710	0.868
	Qwen2.5-14B	0.707	0.898
	Qwen2.5-32B	**0.738**	**0.903**

B Datasets

TREC DL 19-22 are built on the MS MARCO passage collections: TREC-DL 2019–2020 uses MS MARCO V1, while 2021–2022 uses MS MARCO V2, containing approximately 8.8M and 138M passages, respectively. Each query–passage pair is annotated with one of four relevance levels: perfectly relevant (3), highly relevant (2), related (1), or irrelevant (0). Following common practice in TREC-DL studies (Meng et al., 2025; Arabzadeh et al., 2021b), we binarize relevance labels by treating passages with relevance ≥ 2 as positive. The statistics of datasets can be seen in Table 4.

C Training Details.

Ranking Loss. Following (Liu et al., 2025), we employ an importance-aware loss $\mathcal{L}_{\text{rank}}$ to emphasize the learning of top-ranked passages:

$$\mathcal{L}_{\text{rank}} = -\sum_{i \in \mathcal{Y}_{ans}} w_i \log P_\theta(y_i \mid x, y_{<i}), \quad (4)$$

where $\mathcal{Y}_{ans}$ denotes the token indices within the `<answer>` tags. The weight w_i is assigned as:

$$w_i = \begin{cases} 1 + \frac{1}{\log_2(p_i+1)}, & y_i \in \text{passage IDs}, \\ 1, & \text{otherwise}, \end{cases} \quad (5)$$

where p_i represents the rank of the passage corresponding to token y_i in I_j. For non-passage tokens, the weight is set to 1.

Confidence Loss. $\mathcal{L}_{\text{conf}}$ is the standard cross-entropy loss calculated over the tokens within the `<confidence>` tags ($\mathcal{Y}_{conf}$):

$$\mathcal{L}_{\text{conf}} = -\sum_{i \in \mathcal{Y}_{conf}} \log P_\theta(y_i \mid x, y_{<i}). \quad (6)$$

Table 4: Statistics for the TREC 2019–2022 Deep Learning (DL) tracks. Rel. labels represent the average number of passages at each relevance level (0/1/2/3) for each query. Top 100 Rel. labels denote the number of passages at each relevance level within the top-100 passages retrieved by BM25.

Track	# Queries	Rel. labels	Top 100 Rel. labels
DL19	43	120 / 37 / 42 / 16	68 / 12 / 13 / 7
DL20	54	144 / 36 / 19 / 12	76 / 12 / 6 / 6
DL21	53	82 / 58 / 44 / 20	66 / 18 / 11 / 6
DL22	76	3769 / 687 / 606 / 22	76 / 17 / 6 / 2

Table 5: QPP performance of self-consistency and QPP-Gen. "Avg." reports the weighted average over Trec DL19-22, where weights are proportional to the number of queries in each dataset. "–" indicates that all predicted values are 1, so the correlation coefficient cannot be computed.

Metrics	Models	Methods	Spearman ↑					ECE ↓				
			DL19	DL20	DL21	DL22	Avg.	DL19	DL20	DL21	DL22	Avg.
$NDCG_{100}$@10	Llama3.1-8B-Instruct	QPP-Gen	0.248	**0.235**	0.381	0.224	0.268	0.331	0.357	0.338	0.480	0.389
		Self-Consis	**0.471**	0.234	**0.419**	**0.277**	**0.337**	**0.149**	**0.184**	**0.103**	**0.245**	**0.179**
	Qwen2.5-7B-Instruct	QPP-Gen	0.291	**0.355**	0.210	0.297	0.289	0.251	0.176	0.208	**0.183**	0.200
		Self-Consis	**0.350**	0.247	**0.429**	**0.326**	**0.336**	**0.154**	**0.140**	**0.112**	0.244	**0.171**
	Qwen2.5-14B-Instruct	QPP-Gen	**0.470**	0.377	**0.300**	**0.361**	0.371	0.191	0.142	0.223	0.255	0.208
		Self-Consis	0.364	**0.541**	0.253	0.343	**0.373**	**0.119**	**0.088**	**0.142**	**0.240**	**0.158**
	Qwen2.5-32B-Instruct	QPP-Gen	**0.371**	0.214	0.369	0.258	0.295	0.178	0.162	0.215	0.266	0.212
		Self-Consis	0.355	**0.510**	**0.400**	**0.328**	**0.394**	**0.088**	**0.096**	**0.133**	**0.236**	**0.150**
Precision@10	Llama3.1-8B-Instruct	QPP-Gen	0.355	**0.487**	**0.495**	**0.403**	**0.436**	0.428	0.487	0.442	0.579	0.496
		Self-Consis	**0.591**	0.268	0.340	0.364	0.379	**0.108**	**0.148**	**0.091**	**0.263**	**0.166**
	Qwen2.5-7B-Instruct	QPP-Gen	0.275	**0.531**	0.325	**0.438**	0.403	0.240	**0.091**	0.206	**0.113**	**0.154**
		Self-Consis	**0.558**	0.294	**0.448**	0.425	**0.424**	**0.117**	0.132	**0.096**	0.267	0.166
	Qwen2.5-14B-Instruct	QPP-Gen	**0.624**	0.642	**0.361**	**0.569**	**0.548**	0.200	0.200	0.292	0.282	0.249
		Self-Consis	0.415	**0.702**	0.152	0.418	0.423	**0.103**	**0.108**	**0.139**	**0.238**	**0.158**
	Qwen2.5-32B-Instruct	QPP-Gen	**0.552**	0.591	**0.453**	**0.571**	**0.544**	0.195	0.230	0.308	0.257	0.251
		Self-Consis	0.358	**0.624**	0.342	0.467	0.454	**0.118**	**0.171**	**0.123**	**0.235**	**0.171**
MRR	Llama3.1-8B-Instruct	QPP-Gen	-0.073	0.263	**0.243**	**0.220**	0.180	0.142	0.207	0.255	0.434	0.282
		Self-Consis	**0.166**	**0.270**	0.241	0.216	**0.225**	**0.126**	**0.188**	**0.205**	**0.309**	**0.221**
	Qwen2.5-7B-Instruct	QPP-Gen	**0.180**	0.243	**0.263**	**0.316**	**0.260**	0.240	0.242	0.250	**0.293**	0.261
		Self-Consis	0.037	**0.366**	0.139	0.229	0.204	**0.131**	**0.160**	**0.218**	0.310	**0.219**
	Qwen2.5-14B-Instruct	QPP-Gen	-	**0.364**	0.355	**0.456**	-	0.158	0.174	0.159	**0.316**	0.215
		Self-Consis	0.314	0.255	**0.490**	0.213	0.307	**0.086**	**0.139**	**0.121**	0.370	**0.202**
	Qwen2.5-32B-Instruct	QPP-Gen	-0.068	**0.303**	0.363	**0.342**	**0.260**	0.124	**0.192**	0.189	0.344	0.229
		Self-Consis	**0.331**	0.045	**0.411**	0.119	0.210	**0.110**	0.211	**0.142**	**0.276**	**0.197**

Table 6: Ranking performance on TREC DL datasets. The values represent the weighted mean across DL19-22 datasets. Full refers to ranking only. Full+Verb refers to ranking while also outputting confidence.

Metric	Types	Qwen-7B	Qwen-14B	Qwen-32B
$NDCG_{100}$@10	Full	0.496	0.515	**0.555**
	Full+Verb	**0.509**	**0.516**	0.542

Detailed Training Settings. For listwise methods—Verb-Num, Verb-List, BERTQPP, and qppbert-PL—training effectiveness is ensured by performing label balancing across DCG@10 ranges, resulting in 20k queries for training (see the *Used* row of Table 7). The confidence loss weight is set to $\lambda = 100$ for Verb-Num and $\lambda = 10$ for Verb-List. Training is conducted using the AdamW optimizer with a batch size of 32 and an initial learning rate of 5×10^{-6}. For QPP-Gen-SFT, we follow (Meng et al., 2025) and adopt LoRA with $r = 8$ and $\alpha = 16$. For all trainable baselines, we use training labels under two settings: the original MS MARCO-Train annotations and the augmented annotations. Each method is trained for 5 epochs, and the checkpoint with the best Spearman coefficient is selected for evaluation.

Evaluation Datasets. *In-domain.* We use Qwen3-32B to supplement relevance annotations for additional 5,000 MS MARCO queries. To evaluate QPP across different query difficulties, we perform label balancing based on DCG@10 and employ 2,500 queries for in-domain evaluation. *Out-of-domain (OOD).* For OOD evaluation, we use

Table 7: The average number of relevant documents among the top-10 passages in the ranking lists annotated by humans and by Qwen3 on MS MARCO-Train. "Used" denotes the training data actually used after label balancing.

Annotators	Qwen2.5-7B-Instruct		Qwen2.5-7B-Instruct100	
	# Queries	# Rel. Passages	# Queries	# Rel. Passages
Human	40,000	0.387	40,000	0.512
Qwen3	40,000	5.244	40,000	7.002
Qwen3+Hum	40,000	5.273	40,000	7.043
Used	20,000	4.586	20,000	5.051

Table 8: Performance of Verb-Num and Verb-List in the in-domain evaluation using MS MARCO.

Metrics	Qwen2.5-7B-Instruct		Qwen2.5-7B-Instruct100	
	Verb-Num	Verb-List	Verb-Num	Verb-List
DCG@10(Norm)	0.470	0.462	0.551	0.540
Confidence	0.570	0.554	0.634	0.559
Spearman	0.823	0.769	0.766	0.756
ECE	0.102	0.089	0.087	0.064

TREC DL19-22. The distributions of queries with respect to DCG@10 in these datasets are shown in Figure 7. For in-domain evaluation, ranking performance is computed based on the enhanced ground-truth annotations. For OOD evaluation, it is computed directly using the TREC ground-truth annotations. Since Verb-Num and Verb-List can only perform QPP on their own generated ranking lists, it is not possible to make a perfectly controlled comparison between the two methods. In this paper, we directly compare their QPP performance on their respective ranking lists. As the overall ranking performance of Verb-Num and Verb-List is very similar, this comparison is acceptable. For all other methods, QPP is conducted on the ranking list generated by Verb-Num.

C.1 Annotation Augmentation

We leverage Qwen3-32B for relevance annotation because of its strong reasoning capabilities, which enable relatively accurate relevance judgments. At the same time, we avoid using larger models due to computational cost considerations. The annotation results are shown in Table 7. For Qwen2.5-7B-Instruct, the model judged an average of 5.2 passages among the top-10 ranked results as relevant, whereas for Qwen2.5-7B-Instruct100, the average number of relevant passages increased to 7. However, Qwen3-32B can misclassify irrelevant documents as relevant, leading to an inflated number of positive labels, as illustrated in Figure 7. Training on such data causes the model to overestimate ranking performance. To mitigate this issue, we partition all samples into 10 intervals according to normalized DCG@10 values ranging from 0 to 1, and perform sample balancing across these intervals. After balancing, we retain 20,000 samples for training. A comparison of the data distributions before and after balancing is shown in Figure 7.

C.2 Detailed Analysis

In in-domain evaluation, the QPP performance of both Verb-Num and Verb-List is very strong. As shown in Table 8, the Spearman correlation between the predicted scores and true ranking performance exceeds 0.75 for both methods, with the ECE consistently below 0.1. This suggests that the models effectively perceive their own ranking performance. When comparing the two methods, Verb-Num achieves a higher Spearman correlation than Verb-List, but its ECE is slightly worse. Unlike Verb-List, predicting the absolute ranking performance in Verb-Num requires the model to compute DCG@10, which may require more training data. In contrast, Verb-List only needs to perform relevance judgments without calculating the exact score. As seen in the first row, second column of Figure 7, Verb-List closely matches the true score distribution. However, since Verb-List assigns equal weight to all top-10 passages and does not prioritize higher-ranked passages, its Spearman correlation is lower.

As shown in Table 8, both methods still exhibit slight overconfidence in the in-domain setting. However, the overall ECE remains very small. In the OOD setting (See Table 2), both the ECE and Spearman correlation become weaker, although the results are still state-of-the-art (SOTA). To better understand why the models remain slightly

Table 9: QPP performance on Trec DL19-22 datasets. Bold and underline indicate the best and second-best results per ranking model. Statistical significance ($p < 0.05$) is denoted by * for improvement over QPP-Gen-SFT (supervised on MS MARCO) and † for improvement over QPP-Gen (3-32B). Significance for Spearman is calculated via Steiger's Z-test, and for ECE via a paired t-test.

Rankers	Methods	Spearman ↑					ECE ↓				
		DL19	DL20	DL21	DL22	Avg.	DL19	DL20	DL21	DL22	Avg.
Qwen2.5-7B-Instruct	*Unsupervised*										
	QPP-Gen	0.333	0.412	0.291	0.430	0.375	0.241	**0.151**	0.234	0.164	0.192
	QPP-Gen (3-32B)	0.611	0.608	0.365	0.605	0.551	0.161	0.156	0.225	0.187	0.184
	Supervised on MS MARCO										
	BERTQPP	0.135	0.252	0.270	-0.060	0.129	0.419	0.345	0.369	0.133	0.293
	qppBERT-PL	-0.085	0.254	-0.077	0.229	0.103	0.452	0.384	0.305	**0.077**	0.275
	QPP-Gen-SFT	0.496	0.567	0.354	0.509	0.484	0.236	0.260	0.334	0.370	0.310
	Supervised on Augmented MS MARCO										
	BERTQPP	0.414	0.272	0.310	0.432	0.362	0.384	0.312	0.336	0.105	0.262
	qppBERT-PL	0.521	0.650	-0.024	0.196	0.315	0.406	0.339	0.419	0.177	0.316
	QPP-Gen-SFT	0.478	0.410	0.361	0.288	0.370	0.338	0.390	0.437	0.577	0.454
	Verb-Num	**0.636**	**0.713***†	**0.414**	0.617	**0.596**	0.385	0.323	0.328	0.151*	0.278
	Verb-List	0.484	0.699*	0.335	**0.677***†	0.565	**0.141***	0.175*	**0.202***	0.187*	**0.179**

overconfident after training, we further analyze the annotation results and model predictions, as illustrated in Figure 7. We observe that, on TREC, Qwen3-32B generally overestimates the ranking performance compared with human annotations, which in turn leads to the overconfidence observed in Verb-Num and Verb-List. In addition, Verb-List is overall more consistent with Qwen3's judgments, which also explains why its ECE is lower than that of Verb-Num. This issue does not stem from our method itself, but rather from limitations in annotation resources and cost. We believe that with perfectly accurate annotations and sufficient training data, both Verb-Num and Verb-List could achieve better performance. Addressing this limitation represents a promising direction for future work.

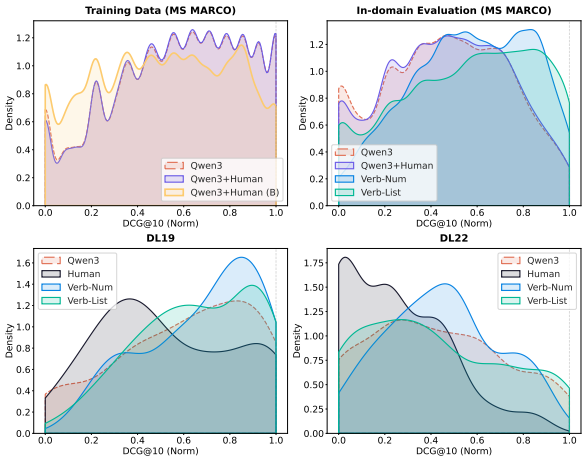


Figure 7: The query distributions corresponding to the predicted DCG@10 scores and the ground-truth scores for different methods on the in-domain and OOD datasets. All results are based on Qwen2.5-7B-Instruct. (B) represents the actual distribution of the training data used after label balancing.

 # Role
You are a search quality rater evaluating the relevance of passages. Your task is to determine if a given passage contains information sufficient to satisfy the query's information need.

 # Task
Perform a binary classification (Output Relevant or Irrelevant).

 # Core Criteria
Label 1 (Relevant - Contains Answer):
The passage MUST contain specific facts that satisfy the query's intent.
- The Litmus Test: Whether the passage contain the answer about the given query? If yes, mark as 1.
- Scope:
  - The passage is dedicated to the query and contains the exact answer.
  - The passage has some answer for the query, but the answer may be a bit unclear, or hidden amongst extraneous information.

Label 0 (Irrelevant - No Answer):
The passage fails to provide the specific answer requested.
- The Litmus Test: If there is no answer for the given query in the passage, mark as 0.
- Scope:
  - The passage seems related to the query but does not answer it.
  - The passage has nothing to do with the query

 # Instructions
You need to carefully discern and not mistakenly classify "The passage seems related to the query but does not answer it" as relevant. Only output whether it is relevant or irrelevant, and do not output anything else.

Query: {Query}
Passage: {Passage}

Figure 8: Instruction for Qwen3-32B Annotation Augmentation.

Instruction: 'Please assess the relevance of the provided passage to the following question. Please output “Relevant” or “Irrelevant”.’
Question: {question}
Passage: {passage}
Output: Relevant/Irrelevant

Figure 9: Instruction for QPP-Gen.

Instruction: I will provide you with 100 passages, each indicated by number identifier []. Rank the passages based on their relevance to the search query: {query}.

[1] Passage 1.
[2] Passage 2.
…
[100] Passage 100.

Search Query: {query}.
Instruction: Rank the 100 passages above based on their relevance to the search query. All the passages should be included and listed using identifiers, in descending order of relevance. The output format should be [] > [], e.g., [4] > [2]. Only response the ranking results, do not say any word or explain.

Figure 10: Instruction for sequence-to-sequence ranking.

Instruction: I will provide you with 100 passages, each indicated by a numerical identifier []. You are asked to:
1.Rank the passages based on their relevance to the search query: {query}.
2.Output your confidence score (0-1) for the top-10 ranking results. This score must reflect the ranking performance like NDCG@10. You should internally grade each passage's relevance (whether it perfectly answers the query), then apply a sharp positional decay where Rank 1-3 are critical. A high score requires 'Highly Relevant' passages to be at the very top; if the top results are only marginally relevant or the ranking is uncertain, the score must be significantly penalized.

[1] Passage 1.
[2] Passage 2.
…
[100] Passage 100.

Search Query: {query}.
Instruction: Rank the 100 passages above based on their relevance to the search query and output your confidence. The output should follow the format <answer></answer><confidence></confidence>, where <answer></answer> contains the ranking results, and <confidence></confidence> contains your confidence. All the passages should be included and listed using identifiers, in descending order of relevance. The output format should be [] > [], e.g., [2] > [1]. Only respond with the ranking results and your confidence, do not say any word or explain.

Figure 11: Instruction for Verb-Num.

Instruction:
I will provide you with 100 passages, each indicated by a numerical identifier []. You are asked to:
1.Rank the passages based on their relevance to the search query: {query}.
2.Output whether the top 10 ranked passages are relevant to the query in the format [1, 0, 1, 0, ...], where 1 indicates relevant and 0 indicates not relevant.

[1] Passage 1.
[2] Passage 2.
…
[100] Passage 100.

Search Query: {query}.
Instruction: Rank the 100 passages above based on their relevance to the search query and output the relevance judgements for the top10 ranked passages (1 for relevant, 0 for not relevant). The output should follow the format <answer></answer><confidence></confidence>, where <answer></answer> contains the ranking results, and <confidence></confidence> contains your relevance judgement results. All the passages should be included and listed using identifiers, in descending order of relevance. The output format should be [] > [], e.g., [2] > [1], Only respond with the ranking results and your judgement results, do not say any word or explain."

Figure 12: Instruction for Verb-List.