AcuRank: Uncertainty-Aware Adaptive Computation for Listwise Reranking

{Soyoung Yoon♣*, Gyuwan Kim♠*}, Gyu-Hwung Cho♣, Seung-won Hwang♣
♣Seoul National University ♠University of California, Santa Barbara
*Equal contribution (author order is randomly determined via coin toss)
SEOUL NATIONAL UNIVERSITY UC SANTA BARBARA

NEURAL INFORMATION PROCESSING SYSTEMS

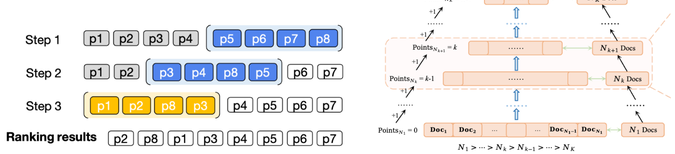
Background and Motivation

- Modern information retrieval pipelines, such as RAG systems, use a first-stage retriever (e.g., BM25 or dense encoder) for recall and speed, followed by a reranker to obtain accurate and reliable relevant documents.
- Recent work has shown **LLM-based listwise rerankers** achieve strong performance by better capturing document interactions while incurring high computational cost because input length limits require multiple calls to cover all candidates.

```
The following are passages related to query {{query}}
[1] {{passage_1}}
[2] {{passage_2}}
(more passages)
Rank these passages based on their relevance to the query.
```

```
[2] > [3] > [1] > [...]
```

- Fixed strategies like sliding windows (left) or tournament-style (right) improve efficiency but lack **adaptivity to query difficulty or document distribution**, motivating an uncertainty-aware framework that dynamically allocates computation.



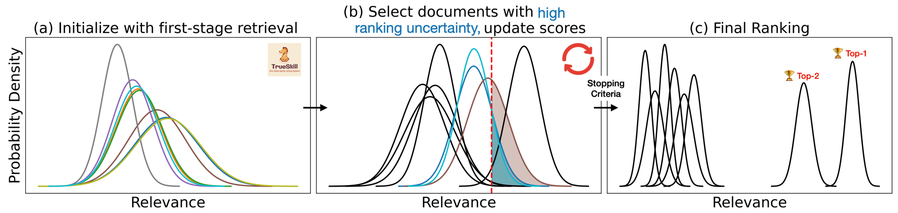
TrueSkill-based Relevance Modeling

- TrueSkill is a principled Bayesian ranking system originally developed for multiplayer games. We recast documents as players, and when one document is ranked above another, it is treated as having won that match.
- We model the latent relevance of each document D_i as a Gaussian variable $x_i \sim \mathcal{N}(\mu_i, \sigma_i^2 + \beta^2)$, where μ_i represents the estimated relevance, σ_i captures epistemic uncertainty, and β represents a global parameter for observation noise.

Ranking Uncertainty Estimation

- The rank of D_i is the number of documents more relevant than D_i under their latent scores. Since each x_i is Gaussian, the exact probability that D_i has rank r can be computed via dynamic programming, but the computation becomes costly and numerically unstable as n grows, due to the need to compute dense integrals and multiply small probabilities.
- To improve scalability, we adopt a more efficient approximation. We define a threshold $t(r)$ such that the expected number of documents whose relevance exceeds $t(r)$ equals the target rank r. Based on the monotonicity of $t(r)$, we efficiently compute $t(r)$ via binary search. We then approximate the cumulative rank probability as $\mathbb{P}(r_i \leq r) \approx \mathbb{P}(x_i > t(r))$, which reflects the chance that x_i exceeds the estimated top-r threshold.

AcuRank Framework



Algorithm 1 AcuRank: Uncertainty-Aware Adaptive Computation for Listwise Reranking

Input: Query q, retrieved documents $\mathcal{D} = \{D_1, \ldots, D_n\}$, listwise reranker g, target rank cutoff k
Output: Ranked list $[D_{r_1} > \cdots > D_{r_n}]$, with top-$k$ used downstream

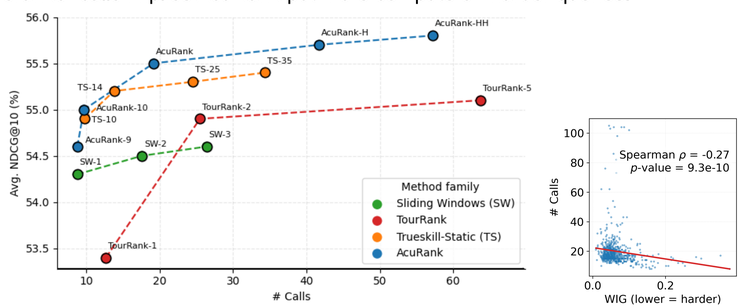
1: Initialize TrueSkill-based relevance scores (μ_i, σ_i) for all $D_i \in \mathcal{D}$
2: **repeat**
3: Select candidate documents $\mathcal{C} \subset \mathcal{D}$ with high ranking uncertainty
4: Partition $\mathcal{C}$ into ordered groups $\{\mathcal{B}_1, \ldots, \mathcal{B}_b\}$
5: Apply listwise reranker g to each $\mathcal{B}_j$ and update TrueSkill scores accordingly
6: **until** $|\mathcal{C}|$ is small, top-k converges, or computational budget is exhausted

Experimental Setup

- Datasets: TREC-DL (DL19, DL20, DL21, DL22, DL23, DL-Hard) and BEIR (TREC-COVID, NFCorpus, Signal-1M, News, Robust04, Touché, DBPedia, SciFact)
- Evaluation metrics
  - Ranking accuracy: Normalized Discounted Cumulative Gain (NDCG@10)
  - Efficiency: the number of of reranker calls per query
- Retrievers: BM25 top-100/1000, SPLADE++ED top-100, Contriever top-100
- Rerankers: RankZephyr, RankGPT (*gpt-4.1-mini*), RankVicuna-7B, Llama-3.3-70B-Instruct
- AcuRank configurations
  - Initialize TrueSkill scores as the mean μ_i to the raw first-stage retrieval score and the standard deviation σ_i to $\mu_i/3$.
  - Select documents whose rank probability $s_i = \mathbb{P}(x_i > t(k))$ falls within the range $(\epsilon, 1-\epsilon)$ with $\epsilon = 0.01$ and $k = 10$.
  - If the number of uncertain documents exceeds the reranker capacity $m = 20$, divide them into equally sized groups using sequential partitioning.
  - Terminate reranking when the number of uncertain documents falls below $\tau = 10$, or the reranker call budget is exhausted.

Experimental Results

- AcuRank consistently lies along the Pareto frontier, achieving stronger accuracy at a given budget or using fewer calls to reach the same target compared to fixed-computation baselines.
- Negative spearman correlation on weighted information gain with respect to reranker calls implies AcuRank put more compute on harder queries.



Experimental Results (Cont.)

- AcuRank maintains strong performance across varying retrieval settings with different first-stage retrievers and reranker model.

Method	TREC-DL DL19	DL20	DL21	DL22	DL23	DL-H	BEIR COVID	NFC	Signal	News	R04	Touche	DBP	Scif	Avg.	# Calls
First-stage retrieval: *BM25 top 1000* | Reranker: *RankZephyr-7B*																
BM25	50.6	48.0	44.6	26.9	26.2	30.4	59.5	32.2	33.0	39.5	40.7	44.2	31.8	67.9	41.1	0.0
SW-1	75.1	78.8	71.5	57.5	49.5	40.9	80.7	38.0	28.9	51.0	48.0	30.9	48.0	76.4	56.2	94.6
TourRank-1	75.4	76.6	71.7	56.6	49.8	42.1	82.7	36.6	29.9	50.9	59.9	33.0	45.5	72.8	56.0	117.1
AcuRank	76.7	75.3	73.1	59.3	53.5	41.0	85.0	37.0	30.7	56.5	63.2	36.2	48.8	76.0	**58.0**	**68.4**

- Each component meaningfully contributes to the effectiveness and the default configuration offers a strong balance between accuracy and efficiency.

Init	Partitioning	Stopping Criterion	TREC	BEIR	All	# Calls
✓	-	-	**59.1**	**52.8**	**55.5**	19.7
×	-	-	59.0	51.7	54.8	**13.4**
✓	Random	-	58.8	52.7	55.3	22.6
✓	-	Top-k stability	58.8	52.4	55.2	22.7

- By processing only the uncertain documents, which can be fewer than the maximum window size, AcuRank uses fewer reranker calls and shorter input lengths, resulting in lower latency and lower FLOPs.

Metric	SW-2 (left) vs. AcuRank (right) DL19	DL20	COVID*	News
NDCG@10 (↑)	**74.6** / 74.2	70.2 / **71.8**	84.4 / **85.3**	52.7 / **53.9**
# Calls (↓)	**18.0** / 18.2	18.0 / **16.3**	27.0 / **21.8**	**18.0** / 18.5
Window size (↓)	20.0 / **16.6**	20.0 / **16.7**	20.0 / **16.3**	20.0 / **16.2**
Input length (↓)	2156 / **1831**	2162 / **1822**	3991 / **3757**	3992 / **3989**
Latency (s) (↓)	126 / **106**	126 / **93**	213 / **142**	179 / **150**
petaFLOPs (↓)	0.61 / **0.52**	0.62 / **0.46**	1.8 / **1.3**	**1.2** / **1.2**

Conclusion

- We propose **AcuRank**, a novel listwise reranking framework that performs adaptive computation guided by uncertainty-aware relevance modeling. Our approach maintains probabilistic relevance estimates using TrueSkill and selectively allocates reranking effort to documents with high ranking uncertainty.
- By focusing computation on ambiguous candidates, AcuRank consistently outperforms fixed-computation baselines and achieve better accuracy-efficiency trade-offs across various retrieval scenarios (with varying reranker models and first-stage retrieval sizes) on different benchmark datasets (TREC-DL and BEIR).