SMMR: Sampling-Based MMR Reranking for Faster, More Diverse, and Balanced Recommendations and Retrieval

Kiryl Liakhnovich
kirill.lyahnovich@gmail.com
T-Bank
Minsk, Belarus

Oleg Lashinin
fotol764@gmail.com
Moscow Institute of Physics and Technology, T-Bank
Moscow, Russian Federation

Andrei Babkin
andrey.babkin.ru71@gmail.com
T-Bank
Moscow, Russian Federation

Michael Pechatov
pechatov@gmail.com
T-Bank
Moscow, Russian Federation

Marina Ananyeva
ananyeva.me@gmail.com
National Research University Higher School of Economics, T-Bank
Moscow, Russian Federation

TL;DR

We propose **Sampled Maximal Marginal Relevance (SMMR)**, a new post-processing reranking method that better balances *relevance* (showing what users like) and *diversity* (exploring new options). Unlike traditional greedy approaches, SMMR introduces controlled randomness, making it more flexible and scalable for large recommendation lists.

Key benefits:

- ⭐ Better trade-off between relevance & diversity
- ⭐ Much faster than greedy methods (**logarithmic speedup**)
- ⭐ Outperforms top baselines in real-world tests
- ⭐ **Open-source** implementation available

Motivation

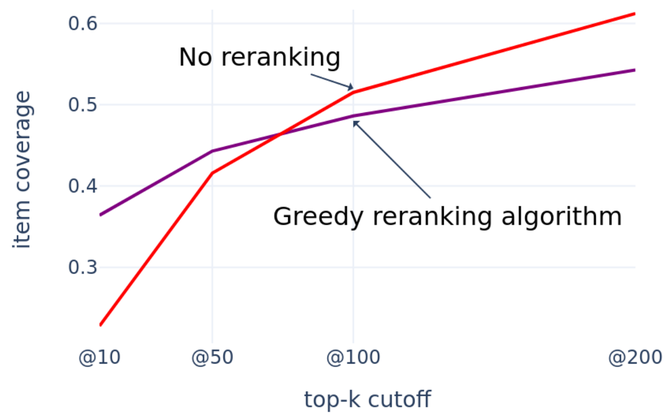
The Problem with Current Methods

Current reranking approaches—such as MMR, DPPs, and SSD—rely on sequential greedy selection. While simple, these methods often converge to suboptimal solutions, failing to effectively balance **relevance** and **diversity**, particularly in large-scale settings with thousands of items. Additionally, their stepwise selection process requires high computational costs when applied to large candidate sets.

The Hidden Cost of Deterministic Selection

Here's the catch: while greedy algorithms *seem* effective for small batches, our research reveals they **hit a breaking point at scale**! Namely, they:

- ✓ **Work well** for 10-20 items, successfully diversifying the retrieval
- ✗ **Fail dramatically** for larger sets, even underperforming *unreranked* results!



Our visualization proves how rigid selection backfires as candidate pools grow. *There has to be a better way… And we found it!*

Methodology & key ideas

💡 Idea #1 - Replace determinism with controlled randomness

Traditional MMR-based reranking selects items iteratively by solving the optimisation problem:

$$i^* = \arg\max_{i \in C \setminus D}(S_i)$$

$$S_i = \lambda \cdot Rel(i) - (1-\lambda) \cdot \max_{j \in D} Sim(i,j)$$

i^*– item to be selected, C– Candidates set, D– Set of already selected items

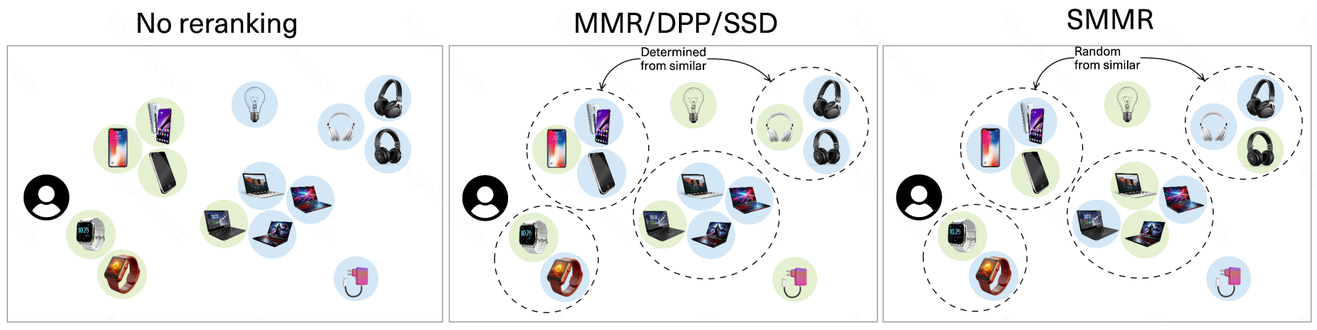
S_i– The score, which balances tradeoff between relevance and diversity

Rel– relevance of item, Sim– similarity function, λ– tradeoff parameter

The idea is to replace deterministic selection with probabilistic sampling

$$i^* \sim P(i^* = i) = \frac{\exp(S_i/t)}{\sum_{j \in C \setminus D} \exp(S_j/t)}$$

t– temperature parameter (controls probability sharpness)



These probabilities are **based on MMR** scores, which means they incorporate the tradeoff between **relevance** and **diversity**. Sampling enables **better exploration** and improves the retrieval diversity!

💡 Idea #2 - Batch selection

To improve efficiency over a one-by-one approach, we introduce dynamic batch selection with **exponentially** increasing sizes. The batch growth rate is controlled by the **scale factor** parameter, which multiplicatively expands the batch size each iteration.

Why Exponentially Increasing Batch Sizes?

- ✔ **Preserve early relevance** – Early batches prioritize high-quality items.
- ✔ **Encourage late-stage exploration** – Later batches trade precision for broader diversity and speed.
- ✔ **Reduce computational cost** – The algorithm requires only *O(log k)* iterations instead of *O(k)*.

The SMMR Algorithm

Input: Candidate set C, Candidate relevances R, Candidate embeddings E, desired set size k
Hyperparameters: Trade-off parameter λ, Temperature t, Scale factor s
Output: Selected set D of size k

Initialize $D \leftarrow \emptyset$, $n \leftarrow 0$;
Get pairwise similarity matrix S_{sim} from embeddings E;
while $|D| < k$ **do**
  foreach $i \in C \setminus D$ **do**
    Compute MMR score:
    $$S(i) \leftarrow \lambda \cdot R(i) - (1-\lambda) \cdot \max_{j \in D} S_{\text{sim}}(i,j)$$
  Compute sampling probabilities with temperature:
  $$P(i) \leftarrow \frac{\exp(S(i)/t)}{\sum_{i' \in C \setminus D} \exp(S(i')/t)}, \quad \forall i \in C \setminus D$$
  Set current batch size: $b \leftarrow \min(s^n, k - |D|)$;
  Sample a batch of b candidates B from $C \setminus D$ using $P(i)$;
  Add sampled items to D: $D \leftarrow D \cup B$;
  Increment $n \leftarrow n + 1$;
return D

Experiments & Evaluation

Metrics

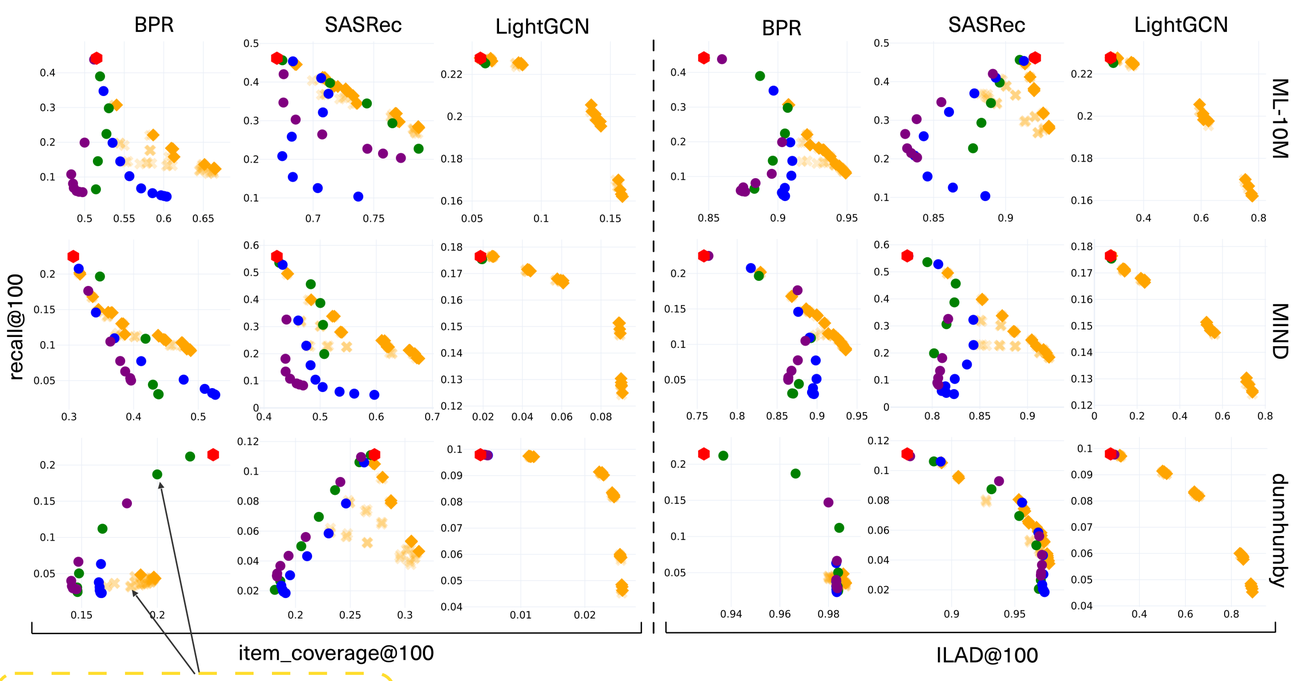
Mean Time@k	Time (seconds) to rerank *k* items (scalability).
Recall@k	Proportion of relevant items retrieved (relevance).
Item Coverage@k	Proportion of unique items covered (diversity).
ILAD@k	Dissimilarity of items across recommendation lists (cross-list diversity).

$$ILAD(L) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(1 - \frac{|L_i \cap L_j|}{|L_i \cup L_j|}\right)$$

📂 Datasets

	Domain	Use Case
MovieLens-10M	Movies	Benchmark for personalized recommendations.
Dunnhumby	Retail purchases	Tests relevance-diversity trade-off in practical scenarios.
MIND	News clicks	Large-scale user behavior for news recommendations.

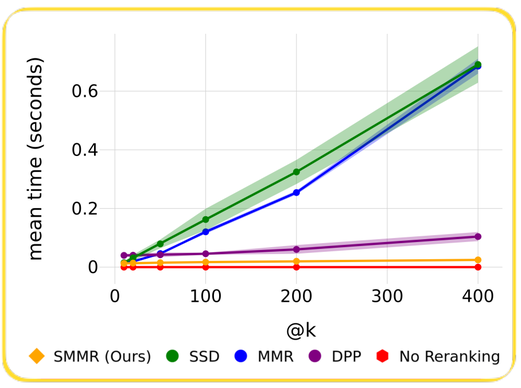
Relevance-diversity tradeoffs compared to other methods



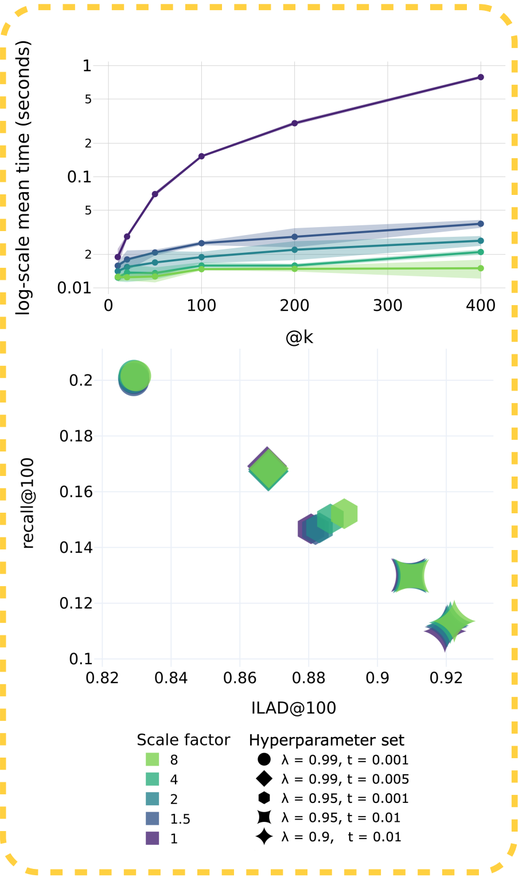
Each point corresponds to a hyperparameter setting from the grid below.
Crosses indicate non-Pareto-optimal points in SMMR's grid search.

Method	Hyperparameters
SMMR	λ: [0.9, 0.95, 0.99], t: [0.001, 0.005, 0.01, 0.03, 0.05], s: [1.5, 2, 4]
MMR	λ: [0.01, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 0.99]
DPP	α: [0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99]
SSD	γ: [10^{-4}, 10^{-2}, 1, 10^2, 10^4, 10^6, 10^8]

🚀 Speed comparison



📈 Scale factor oblations



Impact of sampling introduction

Metric @100	Orig. order	$\lambda = 0.95$		$\lambda = 0.9$	
		MMR	SMMR $t = 10^{-2}$ $s = 1$	MMR	SMMR $t = 10^{-2}$ $s = 1$
ILAD	0.87	0.88 (+0.01)	0.91 (**+0.04**)	0.89 (+0.02)	0.92 (**+0.05**)
Recall	0.22	0.15 (**-0.07**)	0.13 (-0.09)	0.11 (-0.11)	0.11 (-0.11)
Cover.	0.31	0.34 (+0.03)	0.36 (**+0.05**)	0.37 (+0.06)	0.400 (**+0.09**)

🏆 Key Highlights

✨ Smarter Recommendations
- Outperforms traditional methods (MMR/DPP/SSD)
- Better balance between relevant and diverse results
- Naturally enables recommendations refreshing

⚡ Lightning Fast
- Runs in logarithmic time (way faster than competitors)
- Uses smart batch sampling for efficiency
- Perfect for large-scale applications

🔓 Open & Accessible
- Free open-source implementation
- Ready for real-world deployment
- Easy to adopt and extend

Implementation