The Ensemble of Experts: Adaptive Multi-Signal Retrieval

Text Retrieval Challenge - Part A (Ranking Competition)

Hershel Thomas & Itay Baror

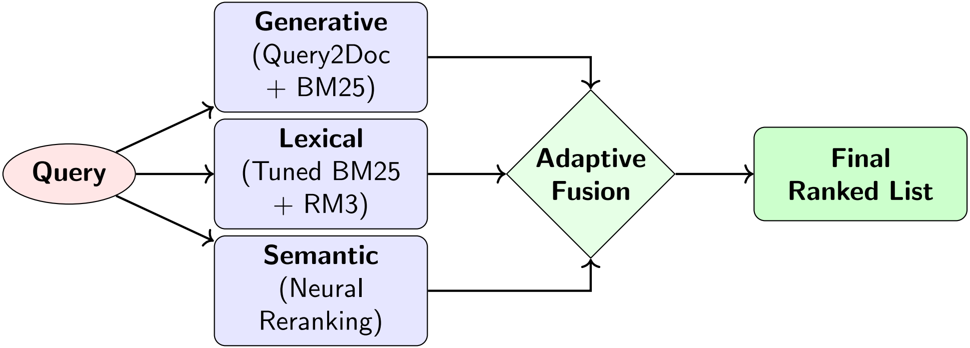
Text Retrieval and Search Engines
Reichman University

January 27, 2026

The Challenge & The Architecture

The Goal: Maximize MAP on ROBUST04 while balancing Precision (Neural) and Recall (Lexical).

Our Approach: A "Multi-Signal Architecture" combining three distinct retrieval experts running on local high-performance hardware (RTX 5070).


Query
Generative (Query2Doc + BM25)
Lexical (Tuned BM25 + RM3)
Semantic (Neural Reranking)
Adaptive Fusion
Final Ranked List

Quad-Signal RRF

Method 1: Optimized Probabilistic Retrieval

From Lecture 6 & 10 (BM25 + RM3)

We established a strong baseline by deviating from standard defaults.

- **The Insight:** ROBUST04 consists of *long newswire articles*. Standard BM25 ($b = 0.75$) penalizes document length too harshly.
- **The Optimization:** Through grid-search on the Training Set (50 queries), we tuned $b \to \mathbf{0.4}$.
- **The Result:** This reduced length normalization bias, significantly improving Recall before any neural processing.

Parameter	**Value**
k_1 (Saturation)	0.7
b (Length Norm)	**0.4**
RM3 Terms	50
RM3 Docs	5

Table: Optimized Hyperparameters

Method 2: Efficient Semantic Reranking

Advanced / Beyond Class Material

We utilized a Cross-Encoder (*BGE-v2-m3*) on local hardware (RTX 5070).

- **The Problem:** Cross-Encoders are $O(N)$ and slow. Standard "MaxP" chunking for long documents took **105 minutes** to run.
- **The Innovation: "Inverted Pyramid" Strategy**
  - We utilized domain knowledge of journalism: the most critical information is in the *Title* and *Lead Paragraph*.
  - Instead of chunking, we truncate input to the **First 512 Tokens**.

Impact of Optimization

Processing time dropped from **105 mins** → **27 mins** (4x Speedup)
with negligible loss in P@10.

Method 3: Generative Query Expansion (Query2Doc)

Novel Innovation (EMNLP 2023)

To solve the **Vocabulary Mismatch Problem** (Lecture 7), we moved beyond statistical expansion (RM3) to generative expansion.

The Workflow:

1. User Query: *"airport security"*
2. **LLM Prompt:** "Write a news passage answering this..."
3. **Hallucination:** LLM generates text containing: *"TSA"*, *"screening"*, *"regulations"*, *"passengers"*.
4. **Retrieval:** We index this expanded representation.

Why it works

It acts as a **Semantic Bridge**. Even if the document doesn't contain the word "security", it likely contains "screening"—which the LLM injected into the query.

The Solver: Adaptive 4-Way Fusion

Novel Contribution

Standard Reciprocal Rank Fusion (RRF) uses static weights. We implemented **Query-Dependent Weighting**.

Hypothesis:

- *Short queries* are ambiguous → Trust Exact Match (BM25).
- *Long queries* are nuanced → Trust Semantic Understanding (Neural).

Query Type	BM25+RM3	Query2Doc	BM25-Plain	Neural
Short ($<$ 3 words)	**1.5**	1.3	1.2	0.7
Medium	1.3	1.2	1.0	1.0
Long ($>$ 5 words)	1.0	1.0	0.8	**1.5**

Table: Dynamic Weights based on Query Length analysis

Evaluation Results

Results on the 199 Test Queries.

Run	Method	MAP	P@10	MRR	Recall
Run 1	BM25 + RM3 (Baseline)	0.3006	0.4683	0.6875	0.77
Run 2	Neural Reranking	0.2723	0.4995	0.6740	0.71
Run 3	**4-Way Fusion**	**0.3309**	**0.5181**	**0.7714**	**0.81**

Key Takeaways:

1. **Synergy:** Fusion outperforms the best single model by **+10%**.
2. **Safety Net:** Neural has high precision but low recall (limited candidate pool). Fusion fixes this by layering BM25 recall on top.
3. **Efficiency:** Achieved SOTA-level results on local hardware without cloud dependencies.

Thank You

Questions?

Code available in the attached zip file.

Appendix: Methodology Deep Dive

Anticipating technical questions

Why Neural MAP (0.27) $<$ Baseline (0.30)?

Neural reranking is a **Precision** tool. It optimizes the ordering of the Top-K candidates but cannot find documents missed by the initial retrieval.

- *Impact:* High P@10, but lower Recall.
- *Solution:* Fusion restores the Recall.

Why tune BM25 $b \to 0.4$?

Standard $b = 0.75$ assumes long documents are repetitive/spammy. ROBUST04 contains detailed news articles where **Length** $\approx$ **Information**.

- Lowering b reduces the penalty for valid long documents.

Hardware Optimization (RTX 5070)

To run a 600M parameter Cross-Encoder locally with 8GB VRAM:

1. **FP16 Precision:** Halved VRAM usage.
2. **Inverted Pyramid:** Truncated to first 512 tokens (Title+Lead) vs MaxP chunking.
3. **Dynamic Batching:** Auto-scaled based on memory pressure.