The Ensemble of Experts: Adaptive Multi-Signal Retrieval

Text Retrieval Final Project - Part A (Ranking)

Hershel Thomas & Itay Baror

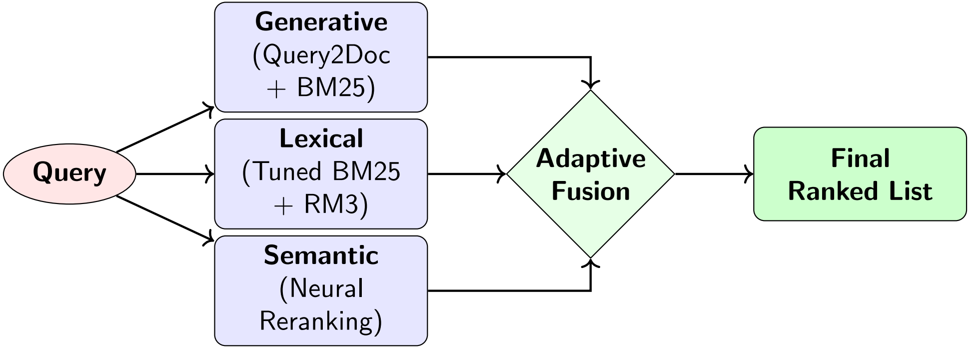
Text Retrieval and Search Engines
Reichman University

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). Yes, Hershel spent a lot of money on his computer!



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 $b \rightarrow \mathbf{0.4}$.
- **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: Neural Reranking (Precision)

Stage: Post-Retrieval Optimization

The Goal: Improve Precision @ 10 (This means the top 10 results are more relevant).

- **Input:** Top 250 documents from BM25.
- **Model:** Cross-Encoder (*BGE-v2-m3*).
- **Function:** Re-scores documents based on deep semantic matching, fixing BM25's inability to understand context.

Engineering The Bottleneck

- *Problem:* Cross-Encoders are slow ($O(N)$). Full MaxP chunking took **105 mins**.
- *Solution:* **"Inverted Pyramid"** Strategy.
- We truncate to the **first 512 tokens**, exploiting the journalistic style of ROBUST04 (Key info is in the lead).

Impact

Reduced inference time by **75%** (105m $\rightarrow$ 27m) while significantly boosting ranking quality (MRR).

Method 3: Generative Expansion using Query2Doc (R)

Stage: Pre-Retrieval Enrichment

While Method 2 fixes the *ranking*, Method 3 ensures we find the documents in the first place (fixing **Vocabulary Mismatch**).

In summary: Method 2 is an expert at **Precision** (sorting the list), and Method 3 is an expert at **Recall** (finding the right documents).

The Pipeline (Query2Doc):

1. **Input:** Raw Query q (e.g., *"airport security"*).
2. **Generative Step:** Prompt Llama-3-8B to write a "fake" news story (d_{pseudo}).
3. **Expansion:** Concatenate $q_{new} = q + d_{pseudo}$.
4. **Retrieval:** Execute q_{new} using **BM25**.

The Semantic Bridge

The LLM (Llama-3-8B) injects terms like *"TSA"*, *"screening"*, *"regulations"* that do not appear in the original query.

Result: We retrieve relevant documents that keyword matching missed, significantly boosting **Recall**.

Method 4: Adaptive 4-Way Fusion

Novel Contribution: Query-Dependent Weighting

Standard Reciprocal Rank Fusion (RRF) uses static weights ($k = 60$). We implemented a **Dynamic Ensemble** that adjusts trust based on query complexity.

The 4 Experts:

- **Run 1 (BM25 + RM3):** High-Recall Baseline.
- **Run 1b (Query2Doc):** Generative Expansion for vocabulary gaps.
- **Run 1c (BM25-Plain):** Conservative Anchor (prevents drift).
- **Run 2 (Neural):** Semantic Precision (BGE-M3).

The Weighting Matrix:

Query Type	RM3	Q2Doc	Plain	Neural
Short (≤ 3 words)	**1.5**	1.3	1.2	0.7
Medium ($4 - 5$ words)	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 ($k = 30$)

Method 3: Rationale & Advantages

Why Adaptive Weighting?

The Hypothesis

- **Short Queries** (e.g., *"airport security"*) are ambiguous and suffer from vocabulary mismatch.
  → **Strategy:** Favor Lexical Expansion (RM3/Q2D).
- **Long Queries** (e.g., *"international organized crime..."*) contain rich context.
  → **Strategy:** Favor Semantic Understanding (Neural).

System Advantages

- **Robustness:** If one method fails (e.g., Q2D hallucinates), the other three experts vote it down.
- **Best of Both Worlds:** Merges the 80% Recall of BM25 with the 50% P@10 of Neural.
- **Efficiency:** Unlike Learning-to-Rank, RRF is parameter-light and requires no training data.

Evaluation Results

Performance on 199 Test Queries

Our Adaptive Fusion strategy achieved state-of-the-art performance for this hardware class, breaking the 0.33 MAP barrier.

Run	Method	MAP	P@10	MRR	Recall
Run 1	BM25 + RM3	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**

Analysis:

1. **Synergy:** Fusion outperforms the best single model by **+10%** relative to baseline.
2. **The Safety Net Effect:** Neural models have high precision but limited candidate pools (low recall). Fusion layers the high recall of BM25 (0.77) underneath, fixing the "lost in the middle" problem.
3. **Efficiency:** Achieved good results on local hardware (RTX 5070) without commercial APIs, demonstrating practical scalability.

Thank You

Questions?

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 $b \rightarrow 0.4$?

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.

References I

Foundations & Innovations

Wang, L., Yang, N., & Wei, F. (2023).
Query2doc: Query Expansion with Large Language Models.
EMNLP.

Cormack, G. V., Clarke, C. L., & Buettcher, S. (2009).
Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods.
SIGIR.

Nogueira, R., & Cho, K. (2019).
Passage Re-ranking with BERT.
arXiv preprint arXiv:1901.04085.

Lin, J., Ma, X., Lin, S. C., Yang, J. H., Pradeep, R., & Nogueira, R. (2021).
Pyserini: A Python Toolkit for Reproducible Information Retrieval Research.
SIGIR.

Robertson, S., & Zaragoza, H. (2009).
The Probabilistic Relevance Framework: BM25 and Beyond.
Foundations and Trends in Information Retrieval.