Optimizing Knowledge Retrieval in Retrieval Augmented Generation

15.C57 - Final Report

Kaibo Zhang
kbzh2558@mit.edu

Annie Liu
yil297@mit.edu

Lauren Zhang
lzhang27@mit.edu

December 5, 2025

1 Introduction

Retrieval-Augmented Generation (RAG) systems pair large language models with relevant information from document collections. Most RAG systems use straightforward retrieval methods, selecting the K documents most similar to the query. However, this approach has weaknesses: retrieved documents often contain redundant information, embedding quality varies unpredictably, and performance depends heavily on embedding choices. Poor retrieval leads to hallucinated or off-topic responses.

This study treats retrieval as an optimization problem rather than a heuristic search task. We showed that standard Top-K retrieval can be reframed as a knapsack problem, then developed three extensions: 1) diversity constraints to prevent selecting similar passages; 2) sparsity penalties that adaptively determine how many documents to retrieve; and 3) robustness mechanisms that account for embedding uncertainty. We built an experimental pipeline that loads embeddings, computes similarity scores, solves mixed-integer optimization models with Gurobi, and evaluates performance on a question-answering dataset. Our experiments compare baseline retrieval, diversity-aware models, norm-based robust models, and cutting-plane methods for sparse adversarial perturbations. The results show when optimization-based methods offer meaningful improvements and interpretations.

2 Data Description

Our experiments use the Mini-Wikipedia RAG dataset [1], which contains 3,200 short Wikipedia-style passages and 918 factual question–answer pairs. The passages span diverse topics, from geography to biology, making the dataset well-suited for testing retrieval methods.

To support our retrieval and optimization experiments, we generate four embedding representations for each document using BERT-base-uncased (768 dimensions), multi-qa-mpnet-base-dot-v1 (768), hkunlp/instructor-large (1024), and intfloat/e5-small-v2 (384). Embeddings are computed in batches, mean-pooled where appropriate, and L2-normalized for cosine similarity search. The question set includes factual verification queries (e.g., ”Was Abraham Lincoln the sixteenth President?”) and entity-specific questions (e.g., ”Did his mother die of pneumonia?”). Each question is embedded using the same four models. These embeddings serve as the core inputs for retrieval and robustness experiments in this study.

3 Methodology

3.1 Heuristic View from Top-k Similarity Search

We begin by formalizing the classical Top-k similarity search as a heuristic baseline for document retrieval in Retrieval-Augmented Generation (RAG). Given a query embedding $q \in \mathbb{R}^d$ and a collection of document embeddings $\{\mu_i\}_{i\in I}$, the heuristic selects the k documents with the highest cosine similarity scores $s_i = \frac{q^\top \mu_i}{\|q\|_2 \|\mu_i\|_2}$. This rule admits a natural optimization interpretation. It is exactly the solution to a binary knapsack problem in which each document has an identical “cost” 1, a total budget of k, and a “value” equal to its similarity score. Letting $x_i \in \{0, 1\}$ denote the selection variable, the formulation is

$$\max_{x\in\{0,1\}^n} \left\{ \sum_{i\in I} s_i x_i \;\middle|\; \sum_{i\in I} x_i = k \right\}.$$

Because all items have identical costs, the knapsack reduces to a deterministic ranking problem: sorting the documents by s_i and choosing the top k yields the unique optimal solution. From an algorithmic perspective, this is equivalent to a k-nearest-neighbor (KNN) query under cosine distance, requiring no combinatorial search beyond sorting. This heuristic, therefore, serves as an interpretable baseline against which more expressive, diversity-aware, or uncertainty-robust retrieval formulations can be systematically developed.

3.2 Variation Across Embedding Models

We evaluate how the retrieved documents change when the *same* query is embedded using different pretrained models. Because each encoder constructs its own vector space—shaped by distinct training corpora, objectives, and inherent representational variance—the induced cosine similarities differ, and thus their Top-k retrieval sets need not overlap. Empirically, the mean pairwise Jaccard distance between models' retrieved sets is 0.8935, indicating that the models select largely non-overlapping documents for identical queries. Table 1 further shows that models such as `multi-qa-mpnet-base-dot-v1` and `hkunlp-instructor-large` achieve stronger semantic alignment (higher F1, cosine similarity, and BERTScore), whereas simpler encoders (e.g., `bert-base-uncased`) perform substantially worse. Accordingly, all subsequent experiments employ the best-performing encoder to ensure consistency and reduce computational cost.

3.3 Enhancement from a Robustness View

Inspired by recent evidence that diversity [2] substantially improves RAG retrieval quality, we extend the Top-k similarity heuristic along three complementary dimensions: (i) incorporating an explicit diversity requirement through a goal-based constraint on the average pairwise cosine among selected documents, (ii) allowing the model to adapt the number of retrieved items via a sparsity penalty rather than fixing k a priori, and (iii) introducing robustness to embedding uncertainty through an adversarial inner problem. This leads to a max–min formulation in which the outer problem selects a subset that balances relevance, diversity, and sparsity, while the inner problem perturbs document embeddings within a user-defined uncertainty set (norm-based or polyhedral) to represent representational variance across encoders. The final robust model

$$\max_{x,y}\ \min_{\tilde{\mu}\in\mathcal{U}} \left\{ \sum_i s_i(\tilde{\mu}_i)x_i - \lambda \sum_i x_i \ \middle|\ \mathrm{McC}(y_{ij}, x_i, x_j)^1,\ \max_{\tilde{\mu}} \sum_{i<j} \cos_{ij}(\tilde{\mu})\, y_{ij} \le \rho_{\mathrm{div}} \sum_{i<j} y_{ij} \right\}.$$

[1] McC denotes the McCormick linearization of the bilinear term $x_i x_j$.

ensures that even under worst-case embedding perturbations, the retrieved set remains diverse, non-redundant, and appropriately sized. This formulation generalizes the heuristic ranking baseline into a principled, uncertainty-aware retrieval model.

4 Results

4.1 Limitations of Standard Uncertainty Models

Norm-based and polyhedral uncertainty sets can fail to preserve the semantic structure needed for reliable retrieval. In a norm-based model, the perturbed embedding is constrained only to lie within a ball of radius ρ, without any restriction on the direction of the perturbation. As illustrated in a toy 3D example (Figure 1), cosine similarity is determined by the angle between vectors, yet a norm ball of sufficiently large radius allows the embedding to rotate almost arbitrarily. This can severely distort the relevance signal and lead the robust solver to suppress documents that are genuinely similar to the query. Pure polyhedral k-sparse uncertainty exhibits a complementary failure mode. Because the adversary is permitted to attack the coordinates that contribute most to the similarity score, the worst-case perturbation often targets precisely the features that encode the true semantic match. In early experiments, even very small values of k caused the retrieved documents to become completely irrelevant, as the strongest coordinates were systematically destroyed. Introducing protection constraints that prevent the adversary from modifying these key coordinates restored meaningful retrieval behavior and yielded more stable and accurate results.

4.2 Answer Quality and Accuracy

We evaluated four uncertainty models for robust retrieval: ℓ_1, ℓ_2, and ℓ_∞ norm-balls, as well as a k-sparse polyhedral set. The retrieved documents were then passed to Llama3, without any specialized prompting or instruction tuning, to generate the final answer for each query (Table 2). The first three yield compact formulations that solve to optimality within seconds and scale well with problem size, whereas the k-sparse model is not tractable in its monolithic form. For this case, we implement a cutting-plane scheme and initialize with the heuristic Top-k solution, which substantially reduces computation time. Across all models, robust variants consistently outperform the baseline heuristic in semantic metrics, with the sparsity penalty λ controlling the trade-off between relevance and redundancy (Table 3, 4, 5, 6). Across all configurations, the robust formulations achieve retrieval quality at least as good as the heuristic Top-k method, and with appropriate hyperparameter tuning can deliver modest but consistent improvements. The optimal sparsity penalty λ tends to remain low, effectively encouraging the model to retain more documents rather than risk discarding relevant ones under worst-case noise.

The behavior of the k-sparse polyhedral uncertainty set highlights the possibility that document retrieval relies on a fundamentally low-dimensional signal structure. Its comparable performance and its failures in early experiments suggest that retrieval depends critically on a small number of "core" coordinates in the dense embedding, while noise in the remaining dimensions can mislead the similarity ranking. In one illustrative example (Table 7), although the correct document is Document 1, Document 2 attains a higher raw cosine score due to noise distributed across non-informative dimensions. This is because Document 1 is much longer in length. The correct answer thus has less accurate representation in the embedding vector. By adversarially erasing only twenty such coordinates, the relationship reverses and the correct document becomes top-ranked. This mechanism is precisely what the protected version of the polyhedral set is designed to capture. It preserves the core semantic signal while allowing perturbations only in directions that do not alter the fundamental relevance structure.

4.3 Consistency and Patterns in Document Retrieval

Using the best-performing hyperparameters identified above, we evaluated robustness across embedding models by computing the Jaccard distance between the solution sets obtained under repeated runs. Among all uncertainty formulations, the ℓ_∞ model achieved the strongest cross-embedding consistency, with an average distance of 0.79. This value is somewhat inflated because different runs may return retrieval sets of unequal size, which mechanically increases the Jaccard distance even when the underlying selections are qualitatively similar. The following example illustrates this behavior (Table 8). For the query "Are beetles endopterygotes?", the robust model consistently centers its selection around document 2394, which explicitly states that "Beetles are endopterygotes with complete metamorphosis." Under the ℓ_1 formulation, each document's similarity is replaced by its worst-case value within the uncertainty set, penalizing embeddings that are unstable or rely on a small number of vulnerable coordinates. As the sparsity penalty λ increases, the model becomes more selective because each document must contribute at least λ units of worst-case relevance to justify inclusion. At lower values of λ multiple documents remain in the set, but once λ reaches 0.65 or 0.72 all but document 2394 fall below this threshold. This mirrors the Jaccard-based consistency findings: the robust formulations, and ℓ_∞ in particular, identify documents whose relevance persists under perturbations, producing retrieval sets that remain stable across embedding spaces and repeated runs.

5 Discussion and Conclusion

This work shows that viewing retrieval as an optimization problem provides a compact and effective alternative to heuristic Top-K search. Diversity constraints, adaptive sparsity, and robustness to embedding uncertainty each yield retrieval sets that are less redundant and more stable across encoders, while remaining computationally practical. Norm-based uncertainty models, particularly the L_∞ formulation, offered the strongest answer performance by suppressing directions that most distort cosine similarity, whereas the K-sparse polyhedral model revealed that retrieval often depends on only a small subset of informative embedding coordinates. Many remaining dimensions behave as noise introduced by chunking or by the encoder itself, which explains failures where the correct document loses its ranking simply because long chunks dilute the signal. A promising direction is to train an optimization-derived sparse masking vector that identifies which embedding coordinates consistently support relevance. Such a mask would not necessarily mirror the adversarial removals but could target structurally noisy dimensions and improve retrieval consistency. Optimization therefore provides a foundation for better retrieval at inference time and for improving the embedding representations used during training.

A Graphs and Tables

Embedding model	F1 score		Cosine similarity		BERTScore F1	
	Mean	Std	Mean	Std	Mean	Std
`bert-base-uncased`	0.218	0.381	0.654	0.288	0.740	0.121
`hkunlp-instructor-large`	0.324	0.440	0.674	0.294	0.771	0.134
`intfloat-e5-small-v2`	0.277	0.414	0.660	0.292	0.750	0.121
`multi-qa-mpnet-base-dot-v1`	0.332	0.438	0.694	0.288	0.771	0.129

Table 1: Heuristic Retrieval Performance Across Embedding Models

RAG Prompt Template

You are a precise question-answering assistant. Based on the provided documents, answer the question with ONLY the direct answer. Do not include explanations, context, or additional information.

Context Documents:
Document 1: <document text>
Document 2: <document text>
. . .

Question: <user question>

Instructions:
– Provide ONLY the direct answer to the question
– Do NOT add phrases such as “Based on the documents...” or “According to...”
– Do NOT provide explanations or reasoning
– If the answer is a single word, number, or short phrase, return only that
– If the answer requires a sentence, make it as brief as possible

Answer:

Table 2: Prompt Used for Answer Generation via Llama3

Parameter	Cosine	Manhattan	BERT-P	BERT-R	BERT-F1
2	0.8352	9.4174	0.8372	0.8540	0.8436
3	0.8649	9.1492	0.8370	0.8715	0.8517
5	0.8935	8.1751	0.8338	0.8815	0.8565
7	0.8993	7.9964	0.8387	0.8929	0.8647
10	0.9246	7.3131	0.8429	0.8953	0.8680

Table 3: Heuristic Answer Performance by Different k Values

Parameter (ρ_{vec}, λ, ρ_{div})	Cosine	Manhattan	BERT-P	BERT-R	BERT-F1
(0.02, 0.50, 0.8)	0.9053	8.3104	0.8303	0.8746	0.8501
(0.02, 0.50, 0.9)	0.9053	8.3104	0.8303	0.8746	0.8501
(0.02, 0.65, 0.8)	0.8490	8.5434	0.8595	0.8678	0.8619
(0.02, 0.65, 0.9)	0.8490	8.5434	0.8595	0.8678	0.8619
(0.02, 0.72, 0.8)	0.8533	8.2144	0.8701	0.8712	0.8690
(0.02, 0.72, 0.9)	0.8533	8.2144	0.8701	0.8712	0.8690
(0.05, 0.50, 0.8)	0.9053	8.3104	0.8303	0.8746	0.8501
(0.05, 0.50, 0.9)	0.9053	8.3104	0.8303	0.8746	0.8501
(0.05, 0.65, 0.8)	0.8623	8.1979	0.8651	0.8692	0.8653
(0.05, 0.65, 0.9)	0.8515	8.4470	0.8624	0.8682	0.8635
(0.05, 0.72, 0.8)	0.8533	8.2144	0.8701	0.8712	0.8690
(0.05, 0.72, 0.9)	0.8533	8.2144	0.8701	0.8712	0.8690

Table 4: L1 Uncertainty Answer Performance by Different Hyperparameters

Parameter (ρ_{vec}, λ, ρ_{div})	Cosine	Manhattan	BERT-P	BERT-R	BERT-F1
(0.02, 0.50, 0.8)	0.9190	7.8559	0.8353	0.8864	0.8582
(0.02, 0.50, 0.9)	0.9190	7.8559	0.8353	0.8864	0.8582
(0.02, 0.65, 0.8)	0.8516	8.5026	0.8609	0.8657	0.8615
(0.02, 0.65, 0.9)	0.8516	8.5026	0.8609	0.8657	0.8615
(0.02, 0.72, 0.8)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.02, 0.72, 0.9)	0.8533	8.2144	0.8701	0.8712	0.8690
(0.05, 0.50, 0.8)	0.9218	7.4589	0.8403	0.8834	0.8592
(0.05, 0.50, 0.9)	0.9114	7.7555	0.8400	0.8733	0.8545
(0.05, 0.65, 0.8)	0.8533	8.2144	0.8701	0.8712	0.8690
(0.05, 0.65, 0.9)	0.8533	8.2144	0.8701	0.8712	0.8690
(0.05, 0.72, 0.8)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.05, 0.72, 0.9)	0.8551	7.9262	0.8794	0.8768	0.8764

Table 5: L2 Uncertainty Answer Performance by Different Hyperparameters

Parameter (ρ_{vec}, λ, ρ_{div})	Cosine	Manhattan	BERT-P	BERT-R	BERT-F1
(0.02, 0.50, 0.8)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.02, 0.50, 0.9)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.02, 0.65, 0.8)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.02, 0.65, 0.9)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.02, 0.72, 0.8)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.02, 0.72, 0.9)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.05, 0.50, 0.8)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.05, 0.50, 0.9)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.05, 0.65, 0.8)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.05, 0.65, 0.9)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.05, 0.72, 0.8)	0.8551	7.9262	0.8794	0.8768	0.8764
(0.05, 0.72, 0.9)	0.8551	7.9262	0.8794	0.8768	0.8764

Table 6: L_∞ Uncertainty Answer Performance by Different Hyperparameters

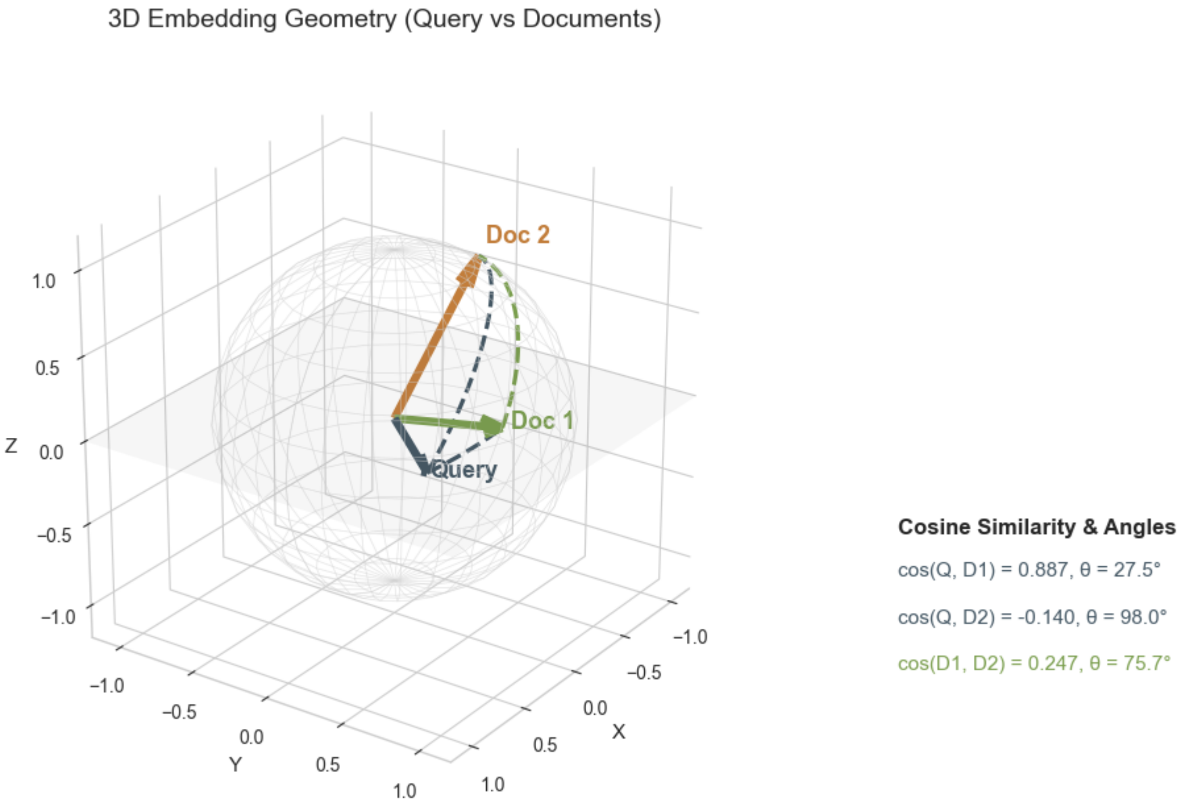

Figure 1: 3-Dimensional Toy Example of Document Retrieval

Doc	Relevant Excerpt	Cosine (orig.)	Cosine (mod.)
1	Abraham Lincoln (...) was the sixteenth President of the United States (...)	0.724	0.751
2	He was the first President to approve a coin, the Lincoln cent (...)	0.747	0.750

Table 7: Toy Example for “Was Abraham Lincoln the sixteenth President of the United States?”

ρ_{vec}	λ	ρ_{div}	Question	Retrieved Document IDs
0.02	0.50	0.8	178	[1010, 1327, ..., ..., 2384, 2385]
0.02	0.65	0.8	178	[2394]
0.02	0.72	0.8	178	[2394]

Table 8: Retrieval Results for “Are beetles endopterygotes?” Under the ℓ_1 Robust Model

References

[1] Hugging Face Community and Contributors. Rag datasets: A collection of small-scale and domain-specific question-answer-passage corpora. https://huggingface.co/rag-datasets, 2024. Accessed: 2025-11-04.

[2] Zhichao Wang, Bin Bi, Yanqi Luo, Sitaram Asur, and Claire Na Cheng. Diversity improves rag: Ranking, clustering, and coverage for retrieval-augmented generation. *arXiv preprint arXiv:2502.09017*, 2025.