Diversity-Aware Retrieval for RAG via Dual Residual Tracking

Kai Roybal
Stanford University
Department of Electrical Engineering
kroybal@stanford.edu

Jacob Rubenstein
Stanford University
Department of Mathematics
jacobr1@stanford.edu

Abstract

Retrieval-Augmented Generation (RAG) systems rely on document retrieval to ground language model responses in external knowledge. However, most retrieval pipelines treat retrieval as a ranking problem, selecting the top-k documents according to similarity scores. While dense retrieval achieves strong performance under ranking metrics, it often returns highly redundant passages that occupy similar semantic regions, wasting limited context window capacity. We investigate retrieval as a coverage-aware optimization problem rather than a pure ranking task. We introduce dual residual tracking, a diversification strategy that iteratively updates dense and sparse query representations after each document selection by removing explained semantic directions and downweighting covered query terms. We compare this approach with other diversification strategies including Maximal Marginal Relevance (MMR), which penalizes candidate documents based on their similarity to the most similar already selected document, and hybrid pricing, which penalizes similarity across the entire selected set. Experiments on the TREC Deep Learning benchmark derived from MS MARCO show that dense retrieval achieves the highest ranking score ($nDCG@k = 0.612$) but produces highly redundant results ($Red@k = 0.770$) for a query. Dual residual tracking reduces redundancy to 0.406 while maintaining competitive ranking performance ($nDCG@k = 0.561$). Diversification also improves set-level retrieval quality: MMR and dual residual tracking achieve higher semantic coverage (0.739, 0.738) than dense retrieval (0.721) and BM25 (0.707), while maintaining strong early retrieval performance ($Hit@3$ up to 0.80). These results show that coverage-aware retrieval strategies can substantially reduce redundancy while preserving ranking performance and improving evidence diversity for RAG systems and query retrieval.

1. Introduction

Large language models have demonstrated strong capabilities across a wide range of natural language processing tasks, but they remain fundamentally limited by the knowledge encoded in their training data in a parametric manner. Because model parameters are static after training, LLMs may produce outdated or incorrect information when answering knowledge intensive questions. Retrieval Augmented Generation (RAG) was introduced to address this limitation by integrating external document retrieval into the generation process, allowing language models to ground their responses in retrieved evidence [9]. In a typical RAG pipeline, a query is encoded into a vector representation and used to retrieve the top-k most relevant documents from a corpus, which are then provided as context to a language model that generates the final answer. While RAG has become a widely adopted architecture for grounding language models in external knowledge, recent work has highlighted several limitations in current implementations [5]. Most systems rely on similarity-based ranking heuristics when selecting documents for retrieval. Dense retrieval models rank documents according to embedding similarity, while sparse methods such as BM25 rely on lexical overlap between the query and candidate documents. Hybrid approaches combine these signals through score fusion or reranking. Although these methods achieve strong performance under standard information retrieval metrics such as nDCG, they implicitly assume that the highest-ranked documents will collectively provide diverse and sufficient evidence for answering a query. In practice, this assumption often fails. Dense retrieval systems frequently produce clusters of highly similar documents that repeat the same information, leading to redundant context that wastes the limited token budget available to the language model. In our experiments, dense retrieval produces redundancy as high as 0.770, indicating that many of the top-ranked passages occupy nearly identical semantic regions while achieving strong ranking performance with $nDCG@k = 0.612$. Sparse retrieval methods exhibit the opposite failure mode, often missing semantically relevant passages when

lexical overlap is weak. Prior work suggests that these limitations arise because ranking-based retrieval optimizes relevance for individual documents rather than the informational value of a document set [19]. As a result, retrieved contexts may contain individually relevant passages that nevertheless provide overlapping evidence and fail to cover all aspects of a query. This issue is particularly important in RAG systems, where language models must reason over a limited number of retrieved passages. These observations suggest that the standard formulation of retrieval in RAG systems may be incomplete. Most current approaches treat retrieval as a ranking problem in which documents are independently scored and the top k results are selected according to similarity. However, the usefulness of a retrieved document depends not only on its relevance to the query but also on the information already contained in previously selected documents. In this work, we investigate retrieval as a resource allocation problem rather than a pure ranking task. Instead of selecting documents solely based on similarity scores, we consider the marginal informational contribution of each candidate document relative to those already retrieved. This perspective aligns retrieval with the objective of RAG systems: constructing a document set that maximizes informational coverage while minimizing redundancy within a limited context window. We evaluate several diversification-based retrieval strategies that incorporate both sparse lexical and dense semantic signals. In addition to traditional ranking metrics, we analyze retrieval behavior using coverage-oriented metrics that better reflect the needs of downstream language model reasoning. Experiments on the TREC Deep Learning benchmark show that diversification substantially reduces redundancy relative to dense retrieval, lowering redundancy from 0.770 to 0.406 in the best coverage-oriented configuration and to approximately 0.20 in more extreme settings. Importantly, these gains do not arise from randomly selecting diverse documents: diversification methods maintain competitive ranking performance, with MMR achieving $nDCG@k = 0.592$ and dual residual tracking achieving 0.561 compared to 0.612 for dense retrieval. At the same time, semantic coverage slightly improves, increasing from 0.721 under dense retrieval to 0.738 under diversification-based retrieval. These results highlight a tension between ranking-based evaluation and coverage-oriented retrieval objectives, suggesting that improving RAG performance may require both new retrieval algorithms and new evaluation frameworks.

2. Related Work

2.1. Sparse Methods

Traditional sparse retrieval methods like TF-IDF and BM25 have dominated information retrieval due to their computational efficiency and exact matching capabilities. DrQA [3] established the modern two-stage paradigm for open-domain question answering by pairing TF-IDF retrieval with a neural reader for answer extraction. However, DrQA documented severe performance degradation when moving from reading comprehension to open-domain settings, with exact match scores dropping from over 80% to below 40%. This highlighted retrieval quality as a critical bottleneck. Wang et al. [18] improved the reader component through global normalization across multiple passages while continuing to rely on BM25 for retrieval. These works demonstrate that while sparse methods excel at exact term matching and maintain fast query times, they fundamentally struggle with vocabulary mismatch. This limitation motivates the development of methods that can capture semantic similarity beyond lexical overlap.

2.2. Dense Methods

Dense retrieval addresses the vocabulary mismatch problem by encoding queries and documents as continuous vectors in a shared semantic space. ORQA [8] demonstrated that dense retrieval could outperform BM25 on open-domain question answering, achieving 33.3% exact match on Natural Questions, although this required weeks of large-scale pretraining. Subsequent work on Dense Passage Retrieval (DPR) showed that dense and sparse retrieval make complementary errors [6]. For example, BM25 can retrieve passages containing salient phrases such as "Thoros of Myr" that dense models miss, while dense retrieval excels at semantic matching, such as linking "body of water" to "Irish Sea".

DPR also explored combining BM25 and dense retrieval through linear score fusion, yielding modest improvements. However, fixed weighting schemes cannot adapt to query characteristics, provide no mechanism for selecting complementary passages, and ignore computational trade-offs between retrieval signals. As a result, simple score fusion does not necessarily produce better document sets. More broadly, both sparse and dense retrieval methods rely on ranking heuristics that approximate relevance at the individual document level. This limitation motivates treating context retrieval as a structured optimization problem that accounts for the overall informational value of the retrieved document set.

2.3. Redundancy and Diversification Methods

When documents are selected purely based on similarity to the query, retrieval pipelines often return clusters of highly similar passages that repeat the same information, reducing the effective coverage of the retrieved context and consuming tokens that could otherwise provide complementary evidence. Zero-RAG [11] examines redundancy from the perspective of overlap between model-internal knowl-

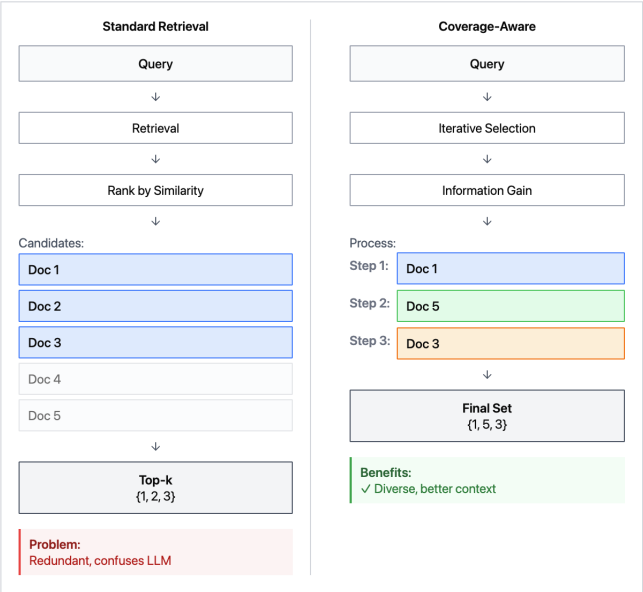


Figure 1. Comparison of standard top-k retrieval (left) versus coverage-aware retrieval (right). Standard retrieval ranks documents by similarity to the query, often selecting redundant documents (Doc 1, 2, 3 shown in blue). Coverage-aware retrieval iteratively selects documents that maximize marginal information gain, resulting in diverse coverage (shown with different colors: blue, green, orange).

edge and external corpora. The authors show that retrieving passages containing knowledge already internalized by the model can degrade performance. Their experiments demonstrate that adding redundant passages to questions the model can already answer reduces accuracy by approximately 20 percentage points. To address this issue, they introduce a mastery-based metric that estimates whether a model already possesses the information contained in a passage and prune documents with high mastery scores. Using this approach, Zero-RAG removes roughly 30% of a Wikipedia corpus while preserving RAG performance and improving retrieval efficiency [11].

While such approaches reduce redundancy at the corpus level, most retrieval systems still rely on ranking heuristics during document selection. These methods operate within ranking frameworks and rely on heuristic trade-offs between relevance and diversity. This limitation motivates approaches that treat retrieval as an explicit optimization problem over document sets rather than independent ranking decisions.

2.4. Combining Sparse and Dense Signal

Hybrid retrieval systems aim to leverage the complementary strengths of dense and sparse query representations. Recent surveys categorize hybrid retrieval approaches into several classes, including parallel score fusion, sequential reranking pipelines, and unified hybrid indexing strategies [12]. In parallel fusion systems, sparse and dense scores are combined through weighted aggregation or reciprocal rank fusion. Sequential pipelines instead use sparse retrieval to generate candidate documents that are subsequently reranked using dense neural models. Unified hybrid systems integrate lexical and semantic representations within a shared retrieval architecture. These approaches have demonstrated improved retrieval effectiveness on benchmarks such as MS MARCO and BEIR by balancing lexical precision with semantic matching capability [12]. In RAG systems, Blended RAG [16] combines BM25 keyword search, dense vector retrieval, and sparse encoder-based semantic search within hybrid query pipelines. Experiments on Natural Questions and TREC-COVID datasets show that hybrid retrievers significantly improve retrieval accuracy compared to single-method approaches, achieving 88.77% top-10 retrieval accuracy on Natural Questions. Similarly, dual-retrieval RAG systems combine dense semantic retrieval with sparse keyword search to improve factual coverage and reduce hallucination in domain-specific tasks such as medical question answering [7]. While these approaches demonstrate the benefits of combining sparse and dense signals, they typically rely on score fusion or multi-stage ranking pipelines rather than explicitly optimizing the final document set.

2.5. Pricing Method

Using convex geometry and pricing-guided discrete selection for retrieval-augmented generation (RAG), Sridhar [17] introduces a two-stage optimization framework that reframes context retrieval as a certified discrete optimization

procedure. This approach demonstrates that pricing algorithms can retrieve document sets with lower redundancy than traditional ranking-based methods, providing a computationally efficient advancement for RAG systems. However, although Sridhar positions this framework outside of traditional retrieval paradigms, it still relies on a single signal, a dense semantic representation, to guide optimization. While such representations may adequately capture the semantic intent of a query, they can still leave a substantial lexical gap. Consequently, we motivate the development of a hybrid pricing algorithm that jointly optimizes dense and sparse signals, enabling retrieval that captures both semantic meaning and lexical specificity rather than relying on a single signal.

3. Methods

3.1. Dataset

We plan to evaluate our retrieval methods using the MS MARCO passage retrieval benchmark together with the TREC Deep Learning (TREC-DL) query sets. MS MARCO is a large-scale dataset constructed from anonymized Bing search queries and web documents, designed to support research on machine reading comprehension and information retrieval [1]. The passage ranking task in MS MARCO contains approximately 8.8 million passages extracted from web documents, along with queries that were sampled from real user search logs.

For evaluation we use the TREC Deep Learning 2019 and 2020 query sets, which provide a curated benchmark for passage retrieval systems [4]. Unlike the original MS MARCO training data, the TREC-DL benchmark includes graded relevance judgments for each query and document set, with relevance levels ranging from 0 (not relevant) to 3 (highly relevant). These graded labels enable evaluation using ranking metrics such as normalized Discounted Cumulative Gain (nDCG), which rewards systems for retrieving highly relevant documents earlier in the ranking.

To construct the evaluation set used in our experiments, we first load all queries and relevance judgments from the TREC-DL 2019 and 2020 benchmarks. Queries that do not contain any relevant documents in the qrels are removed, ensuring that each query in the dataset has at least one judged relevant passage. After filtering, the resulting dataset contains 97 queries with graded relevance annotations.

Rather than retrieving documents from the full MS MARCO corpus during experimentation, we constructed a controlled candidate document pool consisting of the union of all passages that appear in the relevance judgments for the selected queries. This candidate pool contains both relevant and no relevant documents and provides a tractable environment for comparing retrieval algorithms.

Using this restricted candidate pool allows us to directly compare retrieval strategies without the computational cost of searching the full MS MARCO corpus. However, it also introduces a structural limitation, which is that because the pool consists primarily of judged documents, relevant passages may cluster closely in embedding space.

3.2. Document Representations

To support both semantic and lexical retrieval signals, we represent documents using two complementary representations, which are dense semantic embeddings and sparse lexical representations. These representations enable comparison between retreival methods focusing on semantic meaning, and those focusing on keyword appearances, as well as hybrid approaches that combine both signals.

Dense Representations. For dense retrieval, each passage is encoded as a fixed dimensional embedding vector using a pretrained sentence transformer model [14]. Sentence transformers are designed to produce semantically meaningful embeddings for sentences and short passages, allowing similarity between texts to be computed using vector operations.

Let $q \in \mathbb{R}^d$ denote the embedding of a query and $d_i \in \mathbb{R}^d$ denote the embedding of document i. We compute document relevance using cosine similarity:

$$\text{score}_{\text{dense}}(q, d_i) = \frac{q \cdot d_i}{\|q\|\|d_i\|}$$

Sparse Representations. In addition to dense embeddings, we construct a sparse lexical representation of each passage using the BM25 ranking function [15]. BM25 is a probabilistic information retrieval model that ranks documents according to term frequency and inverse document frequency statistics computed from the corpus.

For a query q containing terms t, the BM25 score for document d is computed as:

$$\text{score}_{\text{sparse}}(q, d) = \sum_{t \in q} IDF(t) \cdot \frac{f(t,d) \cdot (k_1 + 1)}{f(t,d) + k_1 \left(1 - b + b \frac{|d|}{\text{avgdl}}\right)}$$

where $f(t, d)$ denotes the frequency of term t in document d, $|d|$ is the document length, and avgdl is the average document length in the corpus. The parameters k_1 and b control term frequency saturation and length normalization.

Sparse representations capture lexical matching behavior that dense embeddings often fail to model, particularly for rare terms, named entities, and exact phrase matches.

3.3. Retrieval Algorithms

We evaluate several retrieval strategies that combine sparse lexical signals and dense semantic embeddings. All methods operate on the same candidate document pool and select documents from said pool.

Dense Retrieval. The dense baseline ranks documents using cosine similarity between the query embedding q and document embedding d

$$s_{\text{dense}}(q, d) = q \cdot d \tag{1}$$

where both embeddings are normalized to unit length. Documents are retrieved by selecting the top k documents according to this score.

Sparse Retrieval (BM25). The sparse baseline ranks documents using the BM25 scoring function discussed above. Documents are retrieved by selecting the top-k BM25 scores.

Hybrid Score Fusion. To combine sparse and dense signals, we use linear score fusion:

$$s_{\text{fusion}}(d) = \alpha\, s_{\text{dense}}(q, d) + (1 - \alpha)\, s_{\text{sparse}}(q, d) \tag{2}$$

where $\alpha \in [0, 1]$ controls the relative weight of semantic and lexical similarity. Scores are normalized before fusion, and alpha is swept through to determine the optimal setting.

Hybrid Maximal Marginal Relevance (MMR). To reduce redundancy among retrieved documents, we implement a hybrid version of Maximal Marginal Relevance (MMR) [2]. At each step the next document is selected by maximizing

$$s_{\text{MMR}}(d) = \alpha\, r(d) - \lambda \max_{s \in S} \text{sim}(d, s) \tag{3}$$

where $r(d)$ is the hybrid relevance score, S is the set of already selected documents, and $\text{sim}(d, s)$ is cosine similarity between document embeddings. In other words, this method penalizes new documents for the already chosen document they are most close to. This minimizes highly redundant documents from being chosen.

Hybrid Pricing. Pricing-based selection penalizes redundancy across the entire selected set rather than the most similar document. At each step we select

$$s_{\text{price}}(d) = r(d) - \lambda \frac{1}{|S|} \sum_{s \in S} \text{sim}(d, s) \tag{4}$$

where $r(d)$ is the hybrid relevance score and S is the set of selected documents.

Dual Residual Tracking. Instead of directly penalizing document similarity, dual residual tracking updates the query representation after each document selection. Let q_t denote the dense query representation at iteration t and d the selected document embedding. The dense residual update is

$$q_{t+1} = q_t - \eta(q_t \cdot d)d \tag{5}$$

which removes the semantic direction explained by the selected document. In parallel, sparse query weights are reduced for tokens appearing in selected documents, encouraging lexical coverage across different query aspects. Retrieval then proceeds using the updated dense and sparse residual queries.

3.4. Evaluation Metrics

Standard information retrieval metrics evaluate retrieval quality by measuring how well individual documents are ranked by relevance. However, for RAG systems the relevant question is not whether each retrieved document is individually relevant, but whether the retrieved *set* collectively provides sufficient and diverse evidence to answer the query. To capture this distinction, we evaluate methods along ranking metrics, which measure individual document relevance and ordering, but coverage-oriented metrics, which measure the informational breadth of the retrieved set as a whole. This dual evaluation framework is central to our argument that ranking based metrics alone are insufficient for assessing retrieval quality in RAG settings, as a set of highly relevant but redundant documents may score well under nDCG while providing little additional value, and actually confusing and worsening a downstream language model.

The metrics used in our experiments are as follows.

nDCG@k (Normalized Discounted Cumulative Gain). For a ranked list of documents up to position k,

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$

where rel_i denotes the relevance score of the document at rank i. The normalized score is computed as

$$nDCG@k = \frac{DCG@k}{IDCG@k}$$

where $IDCG@k$ is the ideal DCG obtained from the optimal ranking. This metric evaluates ranking quality by rewarding highly relevant documents appearing near the top of the list.

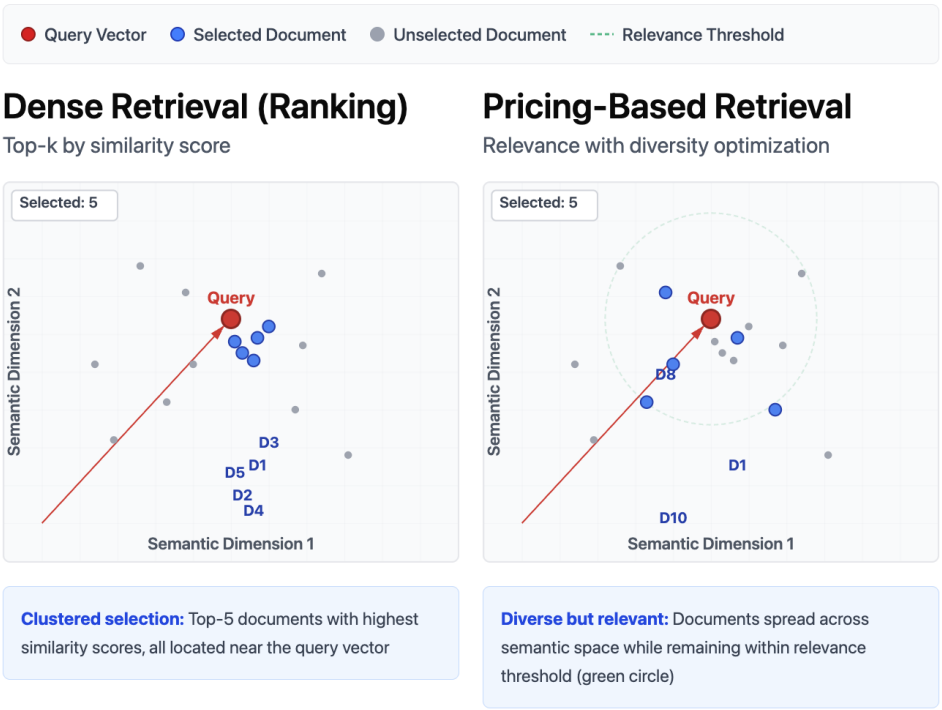


Figure 2. Ranking methods often select documents clustered around the highest similarity region, while diversification-based methods aim to cover multiple semantic aspects of the query.

Red.@k (Redundancy). Let $D_k = \{d_1, d_2, \ldots, d_k\}$ denote the top-k retrieved documents. Redundancy is defined as the average pairwise similarity:

$$Red@k = \frac{2}{k(k-1)} \sum_{i<j} sim(d_i, d_j)$$

where $sim(d_i, d_j)$ represents cosine similarity between document embeddings. Higher values indicate greater redundancy among retrieved documents.

Hit@3. Hit@3 measures whether a relevant document appears within the first three results:

$$Hit@3 = \begin{cases} 1 & \text{at least one relevant doc in the top 3} \\ 0 & \text{otherwise} \end{cases}$$

Across a set of queries Q, the metric is averaged as

$$Hit@3 = \frac{1}{|Q|} \sum_{q \in Q} hit_q$$

This metric captures retrieval success of the most relevant documents.

SemCov@k (Semantic Coverage). Let S denote the set of semantic aspects of a query and let S_k denote the top-k retrieved documents. Semantic coverage measures the fraction of the query's semantic content captured by the retrieved set.

In practice, we approximate semantic aspects using dense document embeddings. Let R_q denote the set of judged relevant documents for query q. Each relevant document is treated as representing a potential semantic interpretation of the query. Coverage of an aspect $r \in R_q$ is measured by the maximum cosine similarity between that relevant document and any retrieved document:

$$c(r) = \max_{d \in S_k} \cos(e(r), e(d))$$

where $e(\cdot)$ denotes the embedding of a document. The semantic coverage score is then computed as

$$\text{SemCov@}k = \frac{1}{|R_q|} \sum_{r \in R_q} c(r)$$

This metric rewards retrieved sets that collectively capture multiple semantic interpretations of the query rather than repeatedly retrieving passages from the same semantic region.

SparseCov@k (Lexical Coverage). Let T_q denote the set of query terms for query q. Sparse coverage measures how well the retrieved documents collectively cover the query's lexical content.

$$\text{SparseCov@}k = \frac{\left|\bigcup_{i=1}^{k} \text{terms}(d_i) \cap T_q\right|}{|T_q|}$$

where $\text{terms}(d_i)$ denotes the set of tokens appearing in retrieved document d_i. This metric measures how well the retrieved documents collectively cover the query's lexical terms.

Irr.@k (Irrelevance). Irrelevance measures the fraction of non-relevant documents in the top-k results:

$$Irr@k = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}(rel(d_i) = 0)$$

where $\mathbf{1}$ is the indicator function. Lower values correspond to higher precision in the retrieved set.

4. Results

4.1. Overview

Figure 3 illustrates the core motivation for this work on a representative query. For the query *is caffeine a narcotic*, dense retrieval selects ten documents that cluster tightly around the query vector in embedding space, ignoring the large portion of the relevant document space that lies further away. MMR begins to spread its selections across the space, while dual residual tracking also reaches substantially further, retrieving documents from semantic regions that dense retrieval never explores. Critically, many of the grade-3 documents,shown in orange, are scattered across the space rather than concentrated near the query, meaning that pure similarity-based retrieval systematically misses them, but both dual residual and MMR are able to find at least one grade 3 document. Note that these visualizations project document embeddings from 384 dimensional space into two dimensions using UMAP, so the apparent clustering and separation in the plot understate the true geometric relationships.

We evaluate dense retrieval, BM25, linear fusion, hybrid MMR, hybrid pricing, and dual residual tracking on the frozen TREC-DL query set described in Section 3. Our goal is not only to identify the single best configuration, but to characterize the tradeoffs between ranking quality, redundancy, semantic coverage, lexical coverage, and retrieval efficiency across method families. Table 1 reports an average configuration for each family.

Across all experiments, the most important qualitative result is that traditional retreival methods achieve high redundancy scores without fail. Methods that achieve the highest nDCG are also the most redundant, clustering selections around the query rather than spreading across the

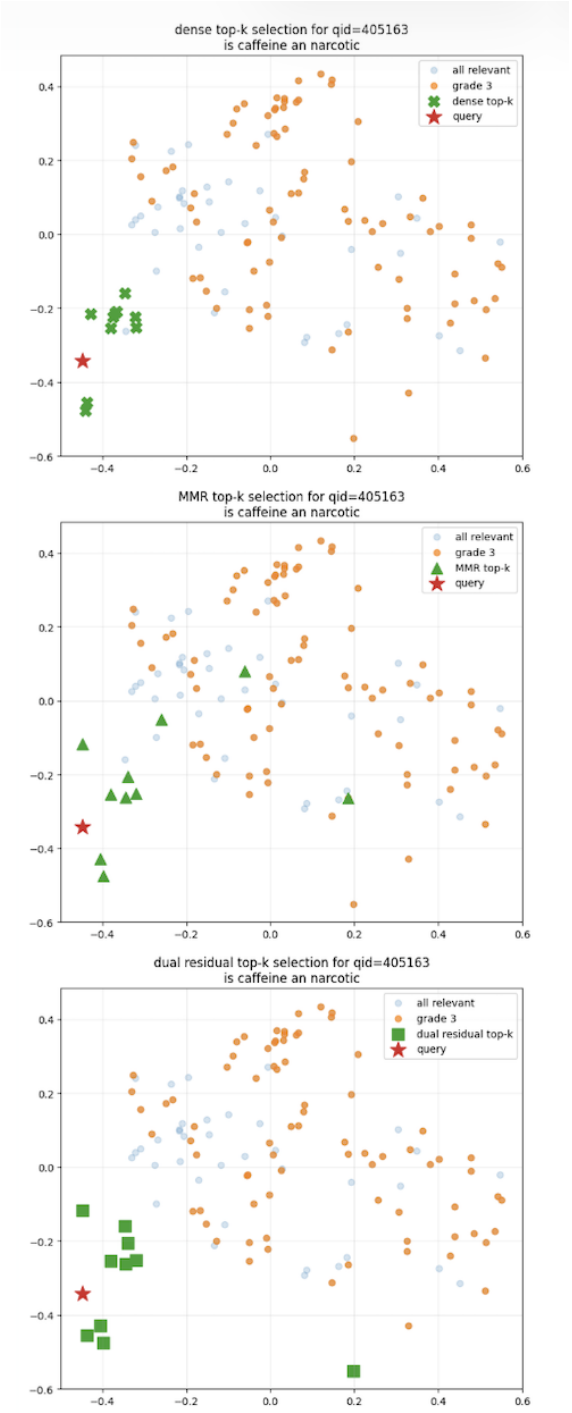


Figure 3. Dense retrieval clusters all selections near the query vector, missing the majority of highly relevant documents scattered across the space. MMR spreads selections more broadly, and dual residual tracking reaches the widest semantic range.

full document space. Diversification and pricing methods intentionally move away from this cluster, improving coverage and reducing redundancy at the cost of lower nDCG, a tradeoff we discuss in detail throughout this section.

4.2. Ranking Performance

We begin with ranking quality under nDCG@k. Dense retrieval and fusion are essentially tied at the top, achieving nDCG@k of 0.612 and 0.613 respectively, with fusion at $\alpha = 0.95$ edging ahead by a negligible margin. This confirms that adding a small lexical component does not hurt ranking quality and can marginally improve it. Among the hybrid methods, pricing reaches 0.604 and MMR 0.602, both remaining competitive with dense retrieval under a ranking oriented evaluation. Although dual residual only reaches 0.586, this still maintains near the same performance as dense. BM25 alone falls well behind at 0.386, which strongly signals that sparse only retrieval is insufficient as a standalone method in this setting.

A revealing signal comes from Hit@3, which once again measures whether at least one grade-3 (most relevant) document appears in the selected documents. Here the ordering reverses noticeably. Dense retrieval scores only 0.722, lower than all pricing methods, while dual residual and hybridpricing both reach 0.773 at their best configurations and MMR reaches 0.763. These differences of $4 - 5\%$ are significant differences. Diversification methods are therefore more reliably surfacing the most relevant documents, even when their overall nDCG is slightly lower. This suggests that dense retrieval's nDCG advantage comes not from finding better documents, but from aggressively clustering selections around the highest-scoring relevance region placing many very similar at least somewhat relevant passages near the top.

Examining redundancy across all configurations reveals that every top-10 nDCG configuration has redundancy above 0.69, which shows that high ranking performance and high redundancy are closely linked, a pattern we examine in detail in Section 4.3.

4.3. Redundancy Reduction

As noted in Section 4.2, high nDCG and high redundancy are closely linked. It has been proven that redundancy in documents during RAG leads to higher rates of hallucination [10]. This section examines redundancy directly and asks whether reducing it comes at the cost of retrieval quality.

Dense retrieval has the highest redundancy of any family at 0.756, fixed across all configurations since it has no mechanism for multiple configurations. Fusion remains similarly clustered, ranging from 0.578 to 0.756 across its alpha sweep, with the balanced configuration sitting at 0.736. MMR and pricing begin to separate from this cluster. The balanced MMR configuration reaches 0.691 and balanced pricing 0.718, both reducing redundancy while maintaining nDCG above 0.587 and 0.604 respectively. BM25 is the least redundant of the fixed methods at 0.523, though this comes alongside a much weaker ranking performance.

Dual residual tracking is qualitatively different from all other families. Across its full parameter sweep, redundancy ranges from 0.201 to 0.695, with a family mean of 0.576. No other method family explores redundancy below 0.430. Its best ranking-oriented configuration achieves redundancy 0.675, comparable to MMR, while its most aggressive coverage-oriented settings push redundancy below 0.21. This behavior follows directly from the residual update mechanism, which forces that once a document is selected, its semantic direction is subtracted from the query representation, making the method structurally biased toward complementary evidence rather than simply penalizing similar candidates as MMR and pricing do.

Figure 4 makes this geometric picture concrete. Each point represents a retrieved document, plotted by its distance to the query vector (x-axis) and its distance to the centroid of the full selected set (y-axis). Dense retrieval produces a tight cluster in the bottom-left corner, all ten selections sit close to both the query and each other, confirming that the method retrieves a highly similar set of passages. MMR spreads its selections further from both the query and the centroid, with distances to the query ranging up to 0.45 and internal spread reaching 0.32. Dual residual tracking reaches substantially further still, with one selection as far as 0.63 from the query and 0.58 from the centroid, a document that dense retrieval would never have retrieved. This geometric spread is the direct consequence of the residual update: each selection shifts the query representation away from already-covered directions, structurally forcing subsequent selections into new regions of the embedding space.

4.4. Coverage vs Redundancy Tradeoff

The previous subsection shows that diversification methods are successful in spreading selections across a wider geometric region than dense retrieval. This section focuses on figure 5, and asks whether that geometric spread translates into better coverage of the relevant document space. The plot reveals two important patterns.First, all non-dual-residual methods cluster tightly in the upper-right corner, achieving high semantic coverage only alongside high redundancy. Second, dual residual's best configurations match or exceed those methods on semantic coverage while sitting at lower redundancy. This begins to answer the question of whether reducing redundancy comes at the cost of coverage quality.

On semantic coverage, the differences across families are modest but meaningful, and they move in the opposite direction to nDCG. Dense retrieval achieves seman-

Family	nDCG@k	Red.@k	Hit@3	SemCov@k	SparseCov@k	Irr.@k	Docs
Dense	0.612	0.756	0.722	0.737	0.871	2.23	10.0
Fusion	0.605	0.733	0.753	0.733	0.905	2.42	10.0
Hybrid MMR	0.587	0.691	0.763	0.753	0.929	2.56	10.0
Hybrid Pricing	0.604	0.718	0.753	0.737	0.899	2.40	10.0
Dual Residual	0.554	0.576	0.763	0.748	0.999	2.84	8.9
BM25	0.386	0.523	0.649	0.692	0.973	4.71	10.0

Table 1. Average configuration per retrieval family.

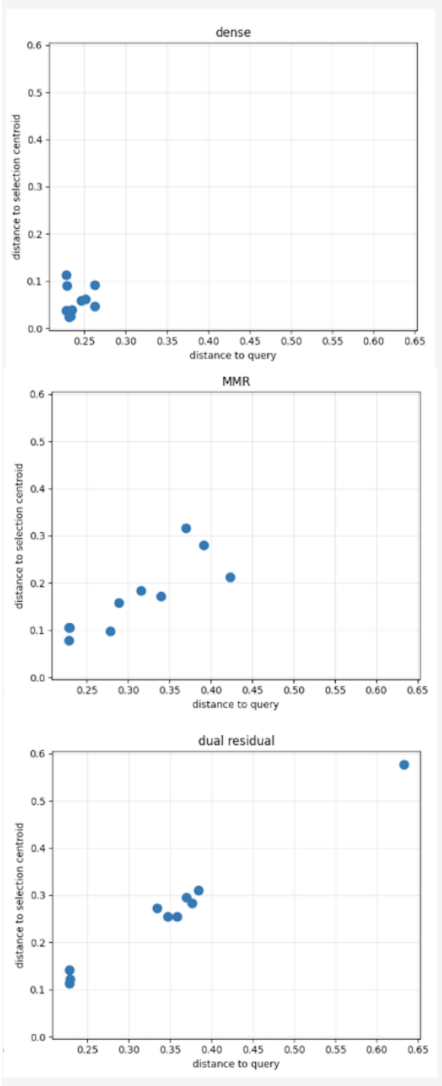


Figure 4. Distance to query vs. distance to selection centroid per retrieved document for three methods on *how does granulation tissue start*. Dense retrieval clusters tightly near the query; MMR and dual residual tracking spread selections progressively further, reflecting greater geometric diversity in the retrieved set.

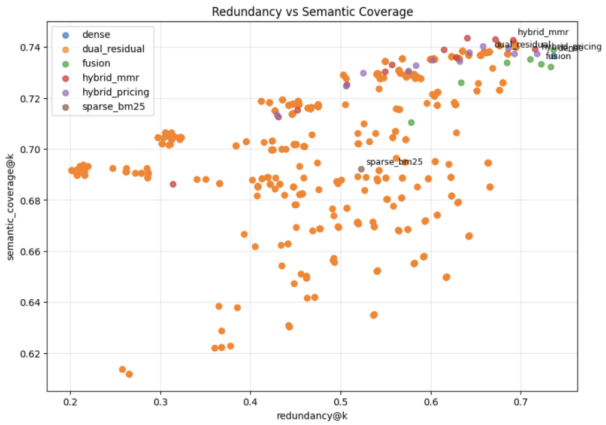


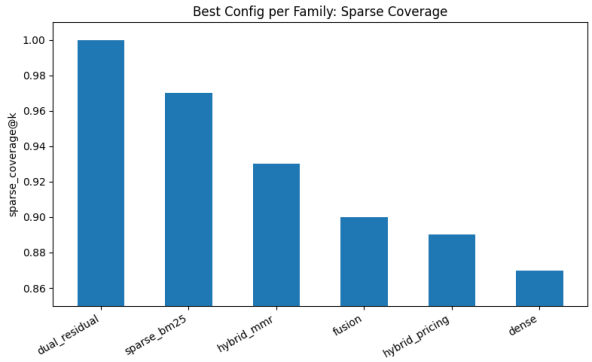
Figure 5. Semantic coverage vs. redundancy across all configurations. Non-dual-residual methods cluster in the upper-right corner, achieving high semantic coverage only alongside high redundancy. Dual residual spans a much wider region, with its best configurations matching other methods on semantic coverage at substantially lower redundancy.

tic coverage of 0.737, fixed across all configurations. Fusion reaches 0.739 at its best, marginally ahead of dense. The clearest gains come from MMR and dual residual, as MMR's best semantic coverage configuration reaches 0.744 at a redundancy of 0.640, and dual residual reaches 0.748 at redundancy 0.693. Critically, both of these best coverage configurations are *less* redundant than dense retrieval, confirming that the redundancy of dense retrieval is not necessary to achieve strong semantic coverage, but is solely a structural artifact of ranking based selection that diversification methods can shed without cost.

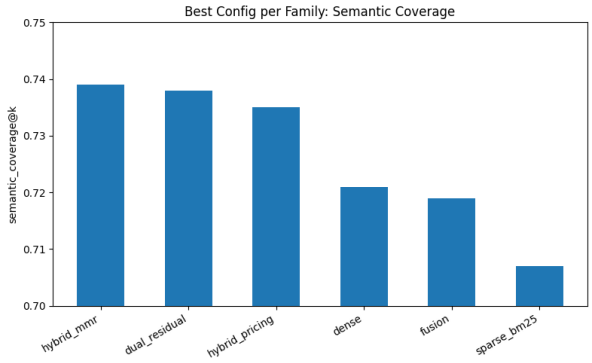
The contrast is far stronger on sparse coverage, where the ordering of methods changes dramatically. Dense retrieval covers only 0.871 of weighted query terms on average, the lowest of any family. This is consistent with its mechanism, as by clustering selections around the highest similarity semantic region, it repeatedly covers the same lexical evidence without seeking out terms present elsewhere in the candidate pool. Fusion improves this to 0.905 in the balanced configuration, and MMR and pricing reach 0.929 and 0.899 respectively. Dual residual is in a different class

entirely. Its *worst* sparse coverage configuration across all 2,592 parameter settings is 0.938, already higher than dense retrieval's best of 0.871. Its balanced configuration reaches 0.999, essentially full query term coverage, while maintaining semantic coverage of 0.738 and nDCG of 0.574. This is the direct consequence of the sparse residual update, which explicitly downweights already covered query terms after each selection, forcing the method to seek out lexical evidence that has not yet been retrieved.

Figure 6 highlights the mismatch between ranking and coverage objectives. Dense retrieval is no longer dominant once the evaluation shifts from ranking to set level coverage. Instead, the best semantic and lexical coverage configurations come from diversification oriented methods, especially dual residual, MMR, and pricing. BM25 remains competitive on sparse coverage because of its lexical bias, but it falls significantly behind on both nDCG and semantic coverage as we have seen.



(a) Sparse coverage



(b) Semantic coverage

Figure 6. Sparse and Semantic Coverage for each Family

4.5. Coverage Efficiency and Early Stopping

Coverage alone is not the full story for RAG systems. A method that achieves high coverage only after retrieving many documents may still be inefficient in practice, especially because LLM's become confused once there is too much context [13]. We therefore evaluate how many documents are required to reach fixed coverage thresholds and whether early stopping reduces the retrieval budget.

Dense retrieval always retrieves the full budget of 10 documents, as do fusion, MMR, pricing, and BM25. Dual residual is the only family that regularly terminates early. Across its parameter sweep, the number of retrieved documents ranges from 2.1 to 10, with a family mean of 8.48. The most aggressive coverage-oriented dual residual configuration retrieves only 6.0 documents on average while maintaining sparse coverage 0.993. This is one of the strongest practical arguments for the residual formulation, as it can concretely reduce the number of retrieved passages without collapsing lexical coverage.

On semantic coverage, the picture is more mixed. Dense retrieval requires 5.60 documents on average to reach 0.80 semantic coverage. Hybrid pricing improves this to 4.88 documents, and the best dual residual configurations reach the same threshold in as few as 1.60 documents, though this occurs alongside a large drop in nDCG to 0.170. Dual residual can therefore be substantially more efficient semantically, but only if one is willing to move well away from ranking-based objectives.

For sparse coverage the gains are clearer and come at a lower cost. MMR and pricing reach 0.80 sparse coverage in as few as 1.74 and 1.62 documents respectively at their best configurations. Dual residual is most efficient of all, reaching the same threshold in just 1.41 documents while maintaining nDCG of 0.527, and 0.90 sparse coverage in 1.75 documents. This is a far more favorable tradeoff than the semantic efficiency configurations. These improvements are consistent with the sparse residual mechanism, because already-covered terms are downweighted after each selection, the method quickly seeks out passages containing uncovered lexical evidence rather than retreading the same terms.

Figure 7 shows that the full-budget methods are concentrated at 10 retrieved documents, whereas dual residual spans a wide horizontal range. The semantic coverage panel suggests that some early-stopped dual residual settings remain competitive with full-budget methods, while the sparse coverage panel shows that even low-budget dual residual configurations often maintain very high lexical coverage. This indicates that the residual stopping rule is not simply truncating retrieval at random; rather, it is terminating after much of the useful lexical evidence has already been collected.

4.6. Main Takeaways

Three conclusions emerge from these results.

First, as expected, dense retrieval remains the strongest baseline under traditional ranking metrics. In this candidate pool, it achieves the highest nDCG and the fewest irrelevant

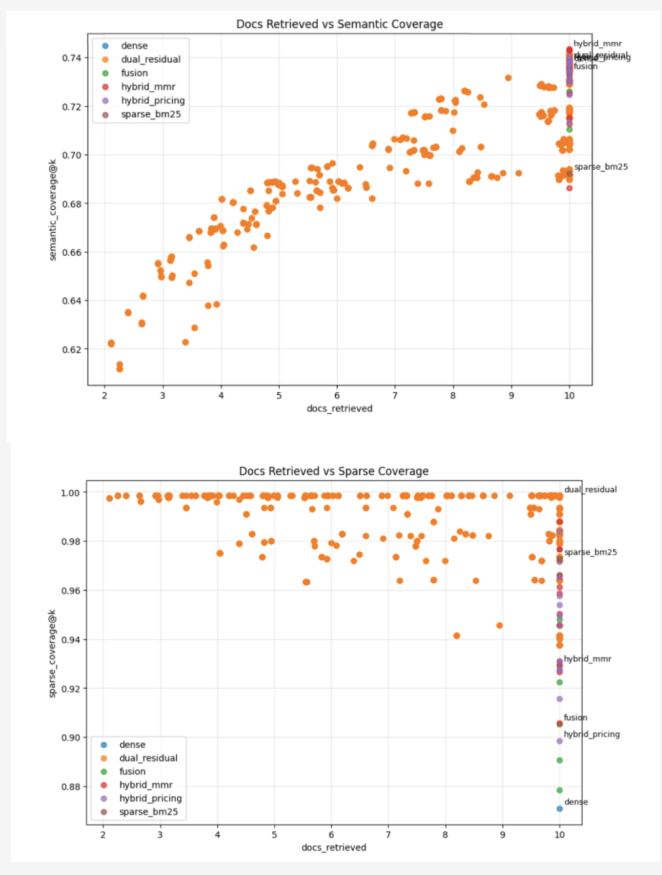


Figure 7. Docs retrieved vs. semantic coverage (top) and sparse coverage (bottom). All non-dual-residual methods are fixed at 10 documents. Dual residual spans the full range; notably, dual residual and pricing methods achieve higher sparse coverage than dense retrieval at full budget while remaining comparable on semantic coverage, suggesting that diversification improves lexical breadth without sacrificing semantic quality.

documents, but it does so by retrieving highly redundant sets of passages.

Second, moderate diversification improves set quality even when it slightly lowers ranking performance. MMR and pricing both reduce redundancy and maintain strong semantic and sparse coverage, suggesting that ranking alone does not fully capture retrieval quality for RAG.

Third, dual residual tracking exposes the clearest trade-offs. Its best ranking oriented configuration remains competitive with the hybrid baselines while improving coverage and reducing redundancy relative to dense retrieval. Its more aggressive configurations can dramatically lower redundancy, improve sparse coverage, and retrieve fewer documents through early stopping, but only at the cost of much lower nDCG. This makes dual residual the most promising family for contexts in which the retrieval budget is limited and evidence diversity matters more than pure ranking performance.

References

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Machine Reading for Question Answering*, 2018. 4

[2] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336. ACM, 1998. 5

[3] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, 2017. Association for Computational Linguistics. 2

[4] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen Voorhees. Overview of the trec 2020 deep learning track. In *Text REtrieval Conference (TREC)*, 2020. 4

[5] Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. A comprehensive survey of retrieval-augmented generation: Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*, 2024. 1

[6] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering, 2020. 2

[7] Jaedong Lee, Hyosoung Cha, Yul Hwangbo, and Wonjoong Cheon. Enhancing large language model reliability: Minimizing hallucinations with dual retrieval-augmented generation based on the latest diabetes guidelines. *Journal of Personalized Medicine*, 14(12):1131, 2024. 3

[8] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering, 2019. 2

[9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[10] Yihan Li, Xiyuan Fu, Ghanshyam Verma, Paul Buitelaar, and Mingming Liu. Mitigating hallucination in large language models (llms): An application-oriented survey on rag, reasoning, and agentic systems, 2025. 8

[11] Qi Luo, Xiaonan Li, Junqi Dai, Shuang Cheng, and Xipeng Qiu. Zero-rag: Towards retrieval-augmented generation with zero redundant knowledge, 2025. 2, 3

[12] Warda Mustafa, Awad Khalil, and Mohamed Attia. Hybrid sparse-dense retrieval: A study of methods, challenges, and recommendations for future research. 1:89–107, 2025. 3

[13] Bhargav Chickmagalur Nanjundappa and Spandan Maaheshwari. Context branching for llm conversations: A version control approach to exploratory programming, 2025. 10

[14] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 4

[15] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. 4

[16] Kunal Sawarkar, Abhilasha Mangal, and Shivam Raj Solanki. Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 155–161, 2024. 3

[17] Anupama Sridhar. Convex geometry + pricing-guided discrete selection for rag (offline segmentation + online reduced-cost retrieval with least-squares certificates. *?*, 2026. 3

[18] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. Multi-passage bert: A globally normalized bert model for open-domain question answering, 2019. 2

[19] Hao Yu et al. Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*, 2024. 2