

SPLADE-COVER: Coverage-Aware Representations for Learned Sparse Retrieval

Siddharth A.K. Singh¹ Eugene Yang² Dawn Lawrie²
Trevor Adriaanse² Andrew Yates²

¹University of Amsterdam

²Johns Hopkins University

s.a.k.singh@uva.nl

eugene.yang@jhu.edu, lawrie@jhu.edu

tadriaa1@jhu.edu, andrew.yates@jhu.edu

Abstract

Long-form retrieval tasks such as report generation, multi-document summarization, and retrieval-augmented generation require systems to surface evidence covering multiple aspects (subtopics) of a complex query. Existing approaches to information coverage typically fall into two families: (1) methods such as Maximum Marginal Relevance (MMR) that rerank initial retrievals using document-similarity heuristics, and (2) expansion-based methods that invoke large language models at query time to produce subqueries, pseudo-answers, or expanded keywords. We propose SPLADE-COVER, a learned sparse retrieval model whose expansion mechanism is trained to be coverage-aware. SPLADE-COVER inherits the SPLADE-v3 architecture but is trained with an objective that distills multi-aspect subquery supervision into a single sparse representation, requiring no language-model invocation or reranking at query time. On benchmarks with nugget-level judgments, SPLADE-COVER often improves over its base model as measured by both information coverage metrics and relevance metrics, at a fraction of the query-time cost of generation-based methods.

1 Introduction

Information coverage is important in retrieval settings where a single query encodes a multi-aspect need that no single document fully satisfies, such as tasks involving long-form question answering, report generation, multi-document summarization, exploratory search, and retrieval-augmented generation over complex queries (Rosset et al., 2025; Ju et al., 2025; Lawrie et al., 2025). For these tasks, retrieving five documents that all address the same aspect (subtopic) is less helpful than retrieving five documents that together cover all aspects.

Existing approaches to first-stage coverage largely fall into two paradigms. *Document similarity for coverage* methods such as Maxi-

mum Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) and its sampling-based variant SMMR (Liakhnovich et al., 2025) reorder an initial retrieval by penalizing documents redundant with those already selected, using inter-document similarity as a proxy for shared topic. By minimizing inter-document similarity in the candidate set while still selecting for relevance, the system surfaces documents that are relevant to different aspects of the main query’s information need. It follows that these methods require no additional query material but are constrained by the first-stage candidate pool: coverage absent from that pool is not recoverable by reranking. *Generation for coverage* methods call large language models at query time to produce additional query material. Some are designed explicitly with information coverage as the objective, such as decomposing a complex query into subqueries whose rankings are fused (Ju et al., 2026); others were originally proposed for relevance improvement through pseudo-documents or expanded keywords (Gao et al., 2023; Wang et al., 2023; Lei et al., 2025; Zhong et al., 2025) but can also be applied to improving coverage. We evaluate baselines from both families. These methods can surface aspect-relevant material that a single query representation would miss, but impose substantial inference cost in LLM calls and additional encoder passes, and vary substantially in coverage effectiveness across datasets.

Both paradigms operate around a first-stage retriever. Learned sparse retrieval (LSR) has emerged as a competitive choice for this stage, combining the efficiency of inverted indexes with the semantic capabilities of pretrained language models (Formal et al., 2021, 2022; Nguyen et al., 2023; Lassance et al., 2024; Nguyen et al., 2026). A key operation in LSR is *expansion*: given a query, the model produces a sparse representation in vocabulary space in which terms not appearing in the original query can receive nonzero weights, addressing the vocab-

ulary mismatch between query and document terms (e.g., surfacing “earbuds” for a query mentioning “headphones”). However, the training objectives behind current LSR models optimize expansion exclusively for relevance: each expanded term should help retrieve documents about the query in isolation. They do not optimize for *information coverage*, the property that retrieved documents collectively address the multiple aspects (subtopics) composing a complex information need (Clarke et al., 2008; Zhai et al., 2003).

We take a third approach that exploits LSR’s expansion behavior to create coverage-aware expansions without using an external LLM expansion model. We instantiate it as **SPLADE-COVER**: a learned sparse retrieval model whose expansion mechanism is trained to be coverage-aware. Rather than correcting for coverage after retrieval or generating additional query material at inference time, SPLADE-COVER distills aspect-level supervision into the expansion head: given a complex query, the model is encouraged to produce a sparse representation whose vocabulary support collectively covers the terms associated with multiple subquery-aspects of that query. SPLADE-COVER inherits the SPLADE-v3 (Formal et al., 2021) architecture and is compatible with document representations produced by SPLADE-v3; it is a drop-in replacement for the SPLADE-v3 query encoder against an existing inverted index, with no LLM invocation at query time. We ask four research questions:

- RQ1:** Does coverage-aware expansion improve information coverage over a relevance-only learned sparse retriever (SPLADE-v3)?
- RQ2:** Does coverage-aware expansion outperform reranking-based coverage methods (MMR, SMMR)?
- RQ3:** Does coverage-aware expansion outperform generation-based coverage methods that call large language models at query time?
- RQ4:** Does coverage-aware expansion preserve effectiveness in a standard relevance-oriented in-domain setting?

We evaluate SPLADE-COVER on three datasets with nugget-level relevance judgments: NeuCLIR24, TREC Interactive, and CRUX-DUC04. SPLADE-COVER improves over SPLADE-v3 and

MMR-style reranking methods while incurring less query latency than generation-based methods.

2 Method

2.1 Coverage-Aware Expansion

SPLADE-COVER is a query encoder trained to produce sparse representations that cover the vocabulary associated with multiple aspects of a complex query, in addition to remaining relevant to the query as a whole.

Let q denote a complex query, expressed as a report request describing an information need with multiple aspects. Let $\{a_1, \dots, a_k\}$ denote a set of aspects associated with q , each expressed as a natural-language subquery that addresses one aspect. The aspects are used only during training as supervision; at inference time only q is provided.

Let f_θ denote the SPLADE-COVER query encoder being trained and let g denote a frozen SPLADE-v3 encoder used as a teacher, identical to the SPLADE-v3 baseline reported in our experiments. We define the aspect-pooled teacher representation as the sparse encoding g produces when the aspects are joined by whitespace and passed as a single input:

$$\bar{s} = g(a_1 \parallel \dots \parallel a_k) \quad (1)$$

This vector contains the union of vocabulary support that SPLADE-v3 would assign to the aspects of q , weighted by their salience under SPLADE-v3’s relevance-trained expansion.

The coverage distillation objective maximizes the cosine similarity (cos) between $f_\theta(q)$ and $\bar{s}$:

$$\mathcal{L}_{\text{cov}} = 1 - \cos(f_\theta(q), \bar{s}) \quad (2)$$

Minimizing $\mathcal{L}_{\text{cov}}$ encourages SPLADE-COVER to expand q into the same coverage-aware vocabulary support that an oracle decomposition would produce, without requiring access to subqueries at inference time.

To preserve the model’s ability to rank documents by relevance, we combine the coverage objective with an InfoNCE contrastive loss (van den Oord et al., 2018) using in-batch negatives:

$$\mathcal{L}_{\text{rel}} = -\log \frac{\exp(f_\theta(q) \cdot f_\theta(d^+)/\tau)}{\sum_i \exp(f_\theta(q) \cdot f_\theta(d_i)/\tau)} \quad (3)$$

where d^+ is a document judged relevant to one of the aspects of q and d_i are in-batch negatives. We use temperature $\tau = 1$ and dot-product similarity.

The number of in-batch negatives that are actually relevant to q in our training data is negligible in practice, so no false-negative correction is applied.

The total loss is a combination:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{cov}} + (1 - \alpha) \mathcal{L}_{\text{rel}} \quad (4)$$

2.2 Model Architecture

Following SPLADE-v3, SPLADE-COVER uses BERT-base (Devlin et al., 2019) as the underlying transformer. Given a query, this base transformer produces contextualized token embeddings $\mathbf{h} \in \mathbb{R}^{L \times 768}$ for L token positions. The dense-to-sparse module applies a projection, layer normalization (LN), and the masked-language modeling (MLM) vocabulary head:

$$\mathbf{s} = \max_t \log(1 + \text{ReLU}(\mathbf{W}_v \cdot \text{LN}(\mathbf{W}_p \mathbf{h}_t))) \quad (5)$$

where $\mathbf{W}_p \in \mathbb{R}^{768 \times 768}$ is a projection to the MLM hidden dimension, $\mathbf{W}_v \in \mathbb{R}^{30522 \times 768}$ is the MLM vocabulary head, ReLU enforces non-negativity, $\log(1 + \cdot)$ compresses dynamic range, and max-pooling aggregates across positions. We initialize all weights from SPLADE-v3. Only the query-side encoder is updated during training; document representations are produced by the original frozen SPLADE-v3 encoder.

2.3 Multi-Aspect Training Supervision

Each training example is a triple of a complex query, an aspect (subquestion) from that query, and a document judged relevant to that aspect. We construct this supervision by adapting Researchy Questions (Rosset et al., 2025). Researchy Questions provides 90,000 complex queries, generative query decompositions, and aspect diversity within the generative decompositions. This made it easier to generate a variety of realistic, synthetic report requests for a RAG task. The labeling process in Researchy Questions derives relevance from the top-level query alone and does not guarantee per-aspect relevance, so we construct fresh per-aspect supervision in three stages.

Synthetic complex queries. We use Llama 3.3 70B to generate complex queries in the form of report requests, conditioning the model on a set of subquestions from each Researchy Questions decomposition. The prompt is shown in Figure 1. The resulting queries are multi-sentence descriptions of information needs whose composition reflects the source subquestions without verbatim repetition.

Prompt: Report Request Generation

Write a description in the first person of an information need that could be answered by reading about some of the following topics. Make up a need related to the topics, but do not repeat any of the topics exactly. However, proper nouns that are entities can be repeated. Where necessary, use a variety of adverbial clauses for narrative flow. The description should only be a few sentences and should not use any numbering or bullets: {}

Figure 1: Prompt for synthetic complex-query (report request) generation in the training data construction pipeline (Section 2.3).

Prompt: Sub-question Generation

Instruction: Write 10 diverse sub-questions that can reveal the information required to answer the given report request. Each sub-question should be self-contained and include the necessary context. Collectively, the sub-questions should fully cover the scope of the report request. Write each sub-question within <q> and </q> tags. Report request: {request} Sub-questions: <q>

Figure 2: Sub-question generation prompt used by the Subquery RRF baseline (Section 3.3) and adapted from LANCER (Ju et al., 2026).

Per-aspect relevance labels. For each subquestion, we retrieve a 100-document pool via BM25, rerank with a Qwen3 0.6B reranker, and apply the CRUX framework (Ju et al., 2025) with Llama 3.3 70B as judge to produce 0–5 relevance grades on the top 20 documents per subquestion.

Filtering and sampling. We retain Researchy Questions decompositions where at least one subquestion has an associated grade-5 document, ensuring each retained complex query has at least one aspect supported by strong relevance signal. For each retained complex query, we sample up to three (subquestion, document) triples to cap the contribution of any single complex query and prevent thematic domination. This produces a training set of 256,649 (complex query, subquestion, document) triples covering 85,732 distinct complex queries and 81,445 unique documents.

3 Experimental Setup

3.1 Datasets

We evaluate on three datasets with nugget-level relevance judgments. CRUX-DUC04 (Ju et al., 2025) contains 50 queries derived from the Document Understanding Conference 2004 summarization data, with an average of 14.2 nuggets per query; queries are taken from the “problem statement” field. The TREC NeuCLIR 2024 Report

Complex query As I reflect on my experience working with minority students, I’ve become increasingly concerned about the long-term effects of being misidentified as requiring special education, particularly because, while I was working with these students, I noticed that they were often being referred to special education at a disproportionately high rate. . . . Because I want to ensure that all students receive an equitable education, I need to understand how the current system affects the self-esteem, identity, and belonging of minority students, and how it perpetuates disparities in the education system.
Header What are the potential causes and factors that contribute to the disproportionality of minority students in special education? Subquestions • How do socioeconomic, cultural, linguistic, and environmental factors affect the identification and referral of minority students to special education? • How do the biases, expectations, and attitudes of teachers, administrators, and evaluators affect the identification and referral of minority students to special education? • How do the availability, quality, and accessibility of general education interventions and resources affect the identification and referral of minority students to special education?
Header What are the implications and consequences of the disproportionality of minority students in special education? Subquestions • How does being in special education affect the academic, social, and emotional outcomes of minority students?
Document Why are certain populations over-represented in special education? One important issue that involves assessment is the high numbers of minority students who are identified for special education services. It is estimated that about twelve percent of the current student population experiences some type of disability. . . .

Figure 3: A training triple (complex query, header questions with subquestions, document) from the SPLADE-COVER training set, drawn from Researchy Questions judgments. This document matches subquestions from two header groups; the model is trained to retrieve it given the complex query and to represent coverage of the headers and their subquestions via the header loss.

Generation Pilot Task (NeuCLIR24) (Lawrie et al., 2025) provides 19 cross-lingual queries with an average of 20.9 nuggets per query, also using the problem-statement field. The TREC 6–8 Interactive Tracks (Over, 1998, 1999; Hersh and Over, 2000) collectively contain 20 queries with an average of 19.8 nuggets per query; we use the concatenation of the title and description fields. All three datasets provide human-annotated nuggets enabling coverage-aware evaluation.

3.2 Metrics

We report two coverage-aware metrics, α -nDCG (Clarke et al., 2008) and subtopic recall (StRecall) (Zhai et al., 2003), at cutoffs 10 and 20. The two capture different aspects of coverage. α -nDCG is rank-sensitive: it discounts a document’s gain by the nuggets it shares with higher-ranked documents, so it is maximized when the ranking supplies new nuggets as early as possible. StRe-

call is rank-insensitive within the cutoff: it measures what fraction of a query’s nuggets appear anywhere in the top k , so a ranking that covers fourteen of nineteen nuggets scores the same regardless of where in the top k those documents fall. A system can therefore improve StRecall without substantially improving α -nDCG by surfacing new nuggets late in the ranking. With 14–21 nuggets per query on average across our datasets, the @5 cutoff is structurally too shallow to measure full coverage and is omitted. We additionally report nDCG at the same cutoffs for relevance, and Recall@100 and Recall@1000 for retrieval depth; these count relevant documents without regard to which nuggets they cover, and so measure the depth of the candidate pool a later stage could draw on rather than coverage itself. NeuCLIR24 uses StRecall at relevance threshold 3; TREC Interactive and CRUX-DUC04 use binary StRecall. Statistical significance is computed via paired t -test against the

SPLADE-v3 baseline at $p < 0.05$ and corrected using Benjamini-Hochberg.

3.3 Baselines

We compare SPLADE-COVER against six baselines covering two first-stage coverage paradigms. **Learned sparse retrieval.** SPLADE-v3 (Lassance et al., 2024), the relevance-only model SPLADE-COVER is finetuned from, provides a reference point for how much coverage-driven training improves over standard LSR.

Reranking-based coverage. MMR (Carbonell and Goldstein, 1998) reranking applied to the top-1,000 SPLADE-v3 retrieval. We grid-search $\lambda \in \{0.01, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 0.99\}$ and rerank depths $\{50, 100\}$. SMMR (Likhnovich et al., 2025) is a sampling-based variant; we additionally grid-search temperatures $\{0.001, 0.005, 0.01, 0.03, 0.05\}$ and scale factors $\{1.5, 2, 4\}$. For each reranking baseline, we report the configuration that performs best across four coverage-aware metrics jointly (α -nDCG@10, α -nDCG@20, StRecall@10, StRecall@20), selecting the configuration with the largest gains over SPLADE-v3 or, where no configuration improves on the baseline, the smallest losses. The selected configurations were MMR: $\lambda = 0.99$, depth= 50; SMMR: $\lambda = 0.99$, temperature= 0.03, scale factor= 1.5, depth= 100.

Generation-based coverage. All four generation-based baselines use Qwen3.5-9B (Qwen Team, 2026) as the LLM, served via vLLM; full sampling parameters and prompt details are reported in Appendix A.

GenQREnsemble (Dhole and Agichtein, 2024) is an ensemble query reformulation baseline that issues multiple prompt variants to the LLM, merges the resulting keywords, and concatenates them with the original query before encoding. AMD-QE (Seo et al., 2025) is a three-agent pseudo-answer expansion baseline that issues 3 calls: a Socratic Questioning Agent generates subquestions, a Dialogic Answering Agent drafts pseudo-answers, and a Reflective Feedback Agent refines them; the final query concatenates the original (repeated three times) with the refined answers. HyDE (Gao et al., 2023) is a pseudo-document expansion baseline that prompts the LLM to generate pseudo-documents per query using task-specific bare-bones templates; each pseudo-document is encoded independently via SPLADE-v3 and the resulting sparse representations are fused by elementwise mean-

pooling. Subquery RRF generates subqueries from a complex query using the Sub-question Generation prompt in Figure 2; each subquery is issued separately to SPLADE-v3 and the resulting rankings are fused via reciprocal rank fusion (Cormack et al., 2009). This paradigm of fusing sub-question rankings for coverage is also explored as a reranking strategy by LANCER (Ju et al., 2026); we adapt this idea here as a query generation baseline. For all four baselines, the expanded query material is encoded by SPLADE-v3. Subquery Oracle issues the ground-truth subqueries associated with each dataset to SPLADE-v3 and fuses results via reciprocal rank fusion (RRF), providing an upper bound on what subquery-fusion can achieve with perfect aspect representations. Ground-truth subqueries are not available for TREC Interactive.

3.4 Training and Inference

We train SPLADE-COVER for one epoch with a batch size of 128, using AdamW with learning rate 5×10^{-5} , linear learning rate decay, no warmup, and zero weight decay. SPLADE-v3’s FLOPS sparsity regularization is not applied during finetuning; sparsity is inherited from the SPLADE-v3 initialization alone. Training was performed on a single NVIDIA A100 GPU. We use random seed 42 throughout (both training and retrieval). At inference we interpolate the SPLADE-COVER query representation with the original SPLADE-v3 representation using a coefficient β on SPLADE-COVER and $1 - \beta$ on SPLADE-v3; this requires running both query encoders, and we report the resulting latency in Section 4. All retrieval uses Anserini (Yang et al., 2018) with random seed 42 to return the top-1,000 documents.

3.5 Hyperparameters

Using the held-out CRUX-MultiNews benchmark as a dev set (Ju et al., 2025), we select SPLADE-COVER’s loss weight α and inference-time interpolation coefficient β . We grid-search $\alpha \in \{0, 0.25, 0.5, 0.75, 1.0\}$, the weighting between the coverage distillation loss and the InfoNCE relevance loss; $\beta \in \{0.25, 0.5, 0.75, 1.0\}$, the weighting between the SPLADE-COVER query representation and the original SPLADE-v3 representation at inference, where $\beta = 1.0$ corresponds to no interpolation. We rank configurations by the number of the eight metrics on which they improve significantly over SPLADE-v3 (paired t -test, $p < 0.05$, Benjamini-Hochberg Correction),

falling back to absolute gains across the same eight metrics only if no configuration attains any significant improvement. Unless otherwise specified, results on SPLADE-COVER use $\alpha = 0.25$ and $\beta = 0.5$ based on this selection procedure.

3.6 Reproducibility

Our training data, training code, baseline query-generation outputs, and synthetic report requests are publicly available at <https://github.com/SiddharthAKSingh/splade-cover>.

4 Results

Table 1 reports SPLADE-COVER against all baselines across three datasets and eight metrics.

RQ1. *Does coverage-aware expansion improve information coverage over a relevance-only learned sparse retriever?* SPLADE-COVER improves over SPLADE-v3 across all three datasets, with statistically significant gains on eight metric-dataset combinations. Two are on coverage-aware metrics: α -nDCG@10 and StR@10 on CRUX-DUC04, two are on relevance ranking: nDCG@10 and nDCG@20 on CRUX-DUC04, and four are on recall: Recall@1000 on NeuCLIR24; Recall@100, and Recall@1000 on TREC Interactive; and Recall@100 on CRUX-DUC04. Relevance metrics typically improve alongside coverage metrics rather than presenting a trade-off. Comparing the full SPLADE-COVER method with a header loss term to the variant with only an InfoNCE loss ($\alpha = 0$) indicates that the header loss contributes to the model’s effectiveness, with the InfoNCE-only variant replicating only 5 of the 8 significant improvements achieved by the full model. Comparing the interpolated SPLADE-COVER method with the variant that does not perform an additional forward pass using the SPLADEv3 encoder ($\beta = 1$), the interpolation-free variant improves over SPLADE-v3 on every metric on NeuCLIR24 and TREC Interactive, but on only half the metrics on CRUX-DUC04; none of these differences reach statistical significance.

RQ2. *Does coverage-aware expansion outperform reranking-based coverage methods?* Neither MMR nor SMMR achieves a statistically significant improvement over SPLADE-v3 on any metric-dataset combination. SPLADE-COVER outperforms both MMR and SMMR in absolute terms on every coverage metric across all three datasets, almost always by substantial margins. This suggests that these

reranking-based implicit diversification approaches are not an effective way to improve information coverage.

RQ3. *Does coverage-aware expansion outperform generation-based coverage methods?* While Subquery RRF attains the strongest absolute coverage on NeuCLIR24 and TREC Interactive, with coverage gains on both (α -nDCG@10/20 and StR@10 on NeuCLIR24, α -nDCG@10/20 on TREC Interactive), none of these gains are statistically significant: Subquery RRF fails to reach significance on any metric-dataset combination due to its relatively high variance. On average subquery RRF outperforms the other generation-based coverage methods. When combined with oracle subqueries, RRF becomes an effective approach, with significant improvements for most metrics on the datasets with oracle subqueries available.

Table 2 shows efficiency metrics for SPLADE-COVER and the baselines: the number of forward passes for the query encoder, the number of LLM calls, and the average query time latency on CRUX-DUC04. Subquery RRF incurs roughly $17\times$ SPLADE-COVER’s per-query latency, and the fastest generation-based baseline has nearly $3\times$ the latency even when its calls are issued concurrently. It is worth noting that the two types of approach are not mutually exclusive: generation-based expansion could be applied on top of SPLADE-COVER.

RQ4 (in-domain analysis). *Does coverage-aware expansion preserve effectiveness in a standard relevance-oriented in-domain setting?* On TREC Deep Learning 2019 (Craswell et al., 2020) and TREC Deep Learning 2020 (Craswell et al., 2021), SPLADE-COVER is comparable to SPLADE-v3 on nDCG@10 on both benchmarks, with no statistically significant difference (Table 3). Neither benchmark is designed to measure information coverage. The changes in effectiveness are comparable to those reported for SoftQE (Pimpalkhute et al., 2024) relative to its base encoders, where training-time distillation of LLM-expanded queries similarly produces no substantial change on these benchmarks. Coverage-aware training therefore does not harm in-domain effectiveness on these standard benchmarks, though it also does not improve effectiveness.

Dataset	System	α -nDCG@10	α -nDCG@20	nDCG@10	nDCG@20	StR@10	StR@20	R@100	R@1000
NeuCLIR24	SPLADE-v3	0.5911	0.6193	0.7991	0.7460	0.7030	0.7878	0.5685	0.8310
	MMR	0.5479	0.5836	0.7947	0.7634	0.6541	0.7640	0.5685	0.8310
	SMMR	0.5477	0.5807	0.7914	0.7618	0.6541	0.7541	0.5685	0.8310
	GenQREnsemble	0.5179	0.5503	0.6868	0.6564	0.6859	0.7707	0.5080	0.8093
	AMD-QE	0.5987	0.6158	0.7467	0.6733	0.7116	0.7636	0.4579	0.7629
	Subquery RRF	0.7052	0.7192	0.8996	0.8144	0.7983	0.8491	0.5735	0.8184
	HyDE	0.6609	0.6975	0.8132	0.7636	0.7385	0.8392	0.5494	0.8017
	SPLADE-COVER	0.5823	0.6166	0.8170	0.7798	0.7082	0.8209	0.5829	0.8498 [†]
	SPLADE-COVER ($\alpha=0$)	0.5838	0.6191	0.8151	0.7857	0.6915	0.8067	0.5717	0.8427
	SPLADE-COVER ($\beta=1$)	0.5936	0.6335	0.8530	0.8010	0.7098	0.8396	0.5734	0.8402
	Subquery Oracle	0.7351 [†]	0.7605 [†]	0.9281 [†]	0.8562 [†]	0.7835 [†]	0.8754 [†]	0.5878	0.8405
TREC Interactive	SPLADE-v3	0.4567	0.4833	0.5558	0.5333	0.3827	0.5048	0.6323	0.8581
	MMR	0.4087	0.4334	0.4943	0.4672	0.3620	0.4914	0.6323	0.8581
	SMMR	0.3990	0.4328	0.4839	0.4675	0.3386	0.4905	0.6323	0.8581
	GenQREnsemble	0.5010	0.5073	0.5841	0.5357	0.4138	0.4891	0.5723	0.8494
	AMD-QE	0.4804	0.4875	0.5384	0.5093	0.4202	0.5012	0.5731	0.8356
	Subquery RRF	0.5477	0.5493	0.6787	0.6106	0.4550	0.5542	0.6728	0.8984
	HyDE	0.5457	0.5490	0.6296	0.5673	0.4499	0.5298	0.5987	0.8607
	SPLADE-COVER	0.4735	0.5041	0.5749	0.5601	0.4177	0.5606	0.6634 [†]	0.9012 [†]
	SPLADE-COVER ($\alpha=0$)	0.4774	0.5044	0.5826	0.5545	0.4189	0.5509	0.6648 [†]	0.8917 [†]
	SPLADE-COVER ($\beta=1$)	0.4658	0.5045	0.5809	0.5787	0.4231	0.5773	0.6859	0.9048
CRUX-DUC04	SPLADE-v3	0.5580	0.6113	0.7019	0.6451	0.6528	0.7941	0.6885	0.8797
	MMR	0.5561	0.6111	0.7025	0.6427	0.6570	0.8011	0.6885	0.8797
	SMMR	0.5564	0.6110	0.7037	0.6427	0.6556	0.8011	0.6885	0.8797
	GenQREnsemble	0.5180	0.5664	0.6435	0.5904	0.6094	0.7419	0.6809	0.8928
	AMD-QE	0.5318	0.5862	0.6490	0.6079	0.6137	0.7695	0.6925	0.8828
	Subquery RRF	0.5477	0.5879	0.6789	0.6169	0.6372	0.7533	0.6952	0.8576
	HyDE	0.5259	0.5722	0.6715	0.6104	0.6057	0.7463	0.6848	0.8419
	SPLADE-COVER	0.5728 [†]	0.6221	0.7241 [†]	0.6640 [†]	0.6685 [†]	0.8063	0.7051 [†]	0.8863
	SPLADE-COVER ($\alpha=0$)	0.5724	0.6220	0.7240 [†]	0.6581 [†]	0.6593	0.7996	0.7027 [†]	0.8839
	SPLADE-COVER ($\beta=1$)	0.5478	0.5986	0.7030	0.6493	0.6524	0.7926	0.7110	0.8784
	Subquery Oracle	0.6989 [†]	0.7340 [†]	0.8479 [†]	0.7790 [†]	0.7764 [†]	0.8835 [†]	0.8539 [†]	0.9710 [†]

Table 1: Results using SPLADE-COVER and several baselines across three datasets. The highest metric value on each dataset, excluding the Subquery oracle upper bound, is shown in **bold**. The [†] symbol indicates scores significantly different from SPLADE-v3 after Benjamini-Hochberg correction ($FDR \leq 0.05$, applied per system across its own 8 metrics $\times$ 3 datasets, $m = 24$), computed via paired t -test on per-query scores. NeuCLIR24 StRecall uses relevance threshold 3; TREC Interactive and CRUX-DUC04 use binary StRecall.

Method	Enc.	LLM Calls	Latency
SPLADE-v3	1	0	380
SPLADE-COVER	2	0	411
GenQREnsemble	1	10	6,069
AMD-QE	1	3	8,206
HyDE	9	1	5,460
Subquery RRF	~ 10	1	6,848

Table 2: Per-query cost on CRUX-DUC04: query-encoder passes, LLM calls, and end-to-end latency (ms) on an A100 against an Anserini index. GenQREnsemble’s ten calls are independent and fall to 1,146 ms if issued concurrently; AMD-QE’s three are sequentially dependent.

Method	DL 2019	DL 2020
Reproduced from Pimpalkhute et al. (2024)		
<i>Dense, without distillation</i>		
DPR	0.6404	0.6281
+ SoftQE	0.6522	0.6380
<i>Dense, distilled from cross-encoders</i>		
SimLM	0.7140	0.6968
+ SoftQE	0.7050	0.7010
E5	0.7252	0.7138
+ SoftQE	0.7282	0.7173
Our results (learned sparse retrieval)		
SPLADE-v3	0.7154	0.7315
SPLADE-COVER	0.7197	0.7342

Table 3: nDCG@10 on TREC Deep Learning 2019 and 2020. Neither difference between SPLADE-COVER and SPLADE-v3 is statistically significant.

5 Related Work

We organize related work around three areas: learned sparse retrieval, the implicit and explicit diversification frameworks for information coverage, and recent expansion-based coverage methods that use large language models at query time.

5.1 Learned Sparse Retrieval

LSR uses neural language models to construct sparse, lexicon-grounded representations of queries and documents, retaining the inverted-index efficiency of traditional bag-of-words methods while

incorporating learned term weighting and vocabulary expansion. This combination has made LSR competitive with dense retrievers across a range of benchmarks (Nguyen et al., 2023; Lassance et al., 2024), supported by efficient sparse index structures (Bruch et al., 2024; Mallia et al., 2024) and by training improvements imported from dense retrieval such as knowledge distillation and hard negative mining (Formal et al., 2022; Nguyen et al., 2023).

SPLADE (Formal et al., 2021, 2022; Lassance et al., 2024) is the state-of-the-art LSR family of architectures that we adapt for our contribution. SPLADE produces a sparse representation over the full vocabulary by passing contextualized embeddings through a masked language modeling head, applying a $\log(1 + \text{ReLU}(\cdot))$ activation, and aggregating across sequence positions via max-pooling. SPLADE-v3 (Lassance et al., 2024) combined KL-divergence and MarginMSE losses, surpassing 0.40 MRR@10 on MS MARCO; we use it as our base model. A defining property of SPLADE’s expansion mechanism is that the resulting representations are explicit and interpretable: each nonzero dimension corresponds to a vocabulary item. SPLADE’s training objectives, however, optimize this expansion exclusively for query-document relevance and sparsity, leaving the expansion mechanism free to encode richer structure that standard training does not target.

5.2 Information Coverage and Diversification

The need to cover multiple aspects of an information need rather than redundantly maximizing relevance was formalized in the search-result diversification literature, motivated by ambiguous, underspecified, and multi-aspect queries (Santos et al., 2010, 2015). Coverage-oriented metrics depart from the Probability Ranking Principle by treating the marginal value of a document as a function of what has already been retrieved. α -nDCG (Clarke et al., 2008) discounts the gain of a document by the subtopics it shares with higher-ranked documents; subtopic recall (StRecall) (Zhai et al., 2003) measures the fraction of subtopics covered at a given cutoff. The concurrently-developed CoverageBench benchmark (Samuel et al., 2026) provides a range of datasets that can be used to evaluate coverage-aware retrieval.

Diversification methods are typically divided into two families (Santos et al., 2010). *Implicit* methods, exemplified by Maximum Marginal Rel-

evance (MMR) (Carbonell and Goldstein, 1998), treat diversity as novelty: each document is scored by a combination of its relevance to the query and its dissimilarity to documents already selected, with a tradeoff parameter λ . Implicit methods require no aspect representation, but the relevance-diversity tradeoff is empirically delicate and the document-similarity proxy is not always a faithful indicator of shared topic. Recent variants include sampling-based MMR (SMMR) (Likhnovich et al., 2025). *Explicit* methods such as xQuAD (Santos et al., 2010) model query aspects directly and score documents by how well they cover aspects not yet satisfied by the ranking. Explicit methods outperform implicit methods when high-quality aspect representations are available (Santos et al., 2015), but they require a source of aspects, which historically has come from query logs, taxonomies, or hand-curation. SPLADE-COVER pursues the same coverage goal but realizes the coverage mechanism inside the learned sparse retriever itself, with coverage-awareness trained into the model rather than enforced at retrieval time.

5.3 Generation-Based Coverage Methods

A recent line of work uses large language models to produce additional query material at retrieval time. HyDE (Gao et al., 2023) generates pseudo-documents that are encoded and used as queries. Query2Doc (Wang et al., 2023) generates plausible answers concatenated with the original query. ReDI (Zhong et al., 2025) and ThinkQE (Lei et al., 2025) iterate query generation and reflection for reasoning-intensive queries. IRCOT (Trivedi et al., 2023) interleaves retrieval with chain-of-thought reasoning steps for multi-hop questions. Subquery decomposition methods such as those in Self-RAG (Asai et al., 2024) and RQ-RAG (Chan et al., 2024) produce explicit subqueries and either rerank or fuse the resulting retrievals, typically using reciprocal rank fusion (Cormack et al., 2009). LANCER (Ju et al., 2026) applies the same paradigm at the reranking stage, aggregating per-sub-question answerability judgments via RRF or coverage-based utility functions. Whether the generated material is concatenated into a single query or issued separately and fused, and whether fusion is performed at retrieval or reranking, is a methodological detail; all of these approaches belong to the same paradigm of generating query material at inference time.

SoftQE (Pimpalkhute et al., 2024) is the closest precedent for our approach: rather than calling an LLM at inference time, it trains a dense query encoder to produce embeddings that approximate those of LLM-expanded queries, distilling the benefit of expansion into the encoder offline. SPLADE-COVER adopts the same training-time distillation strategy but differs in three respects: the underlying retriever is sparse rather than dense, so the distilled signal is vocabulary support rather than embedding direction; the supervision is multi-aspect, with the teacher representation pooled over a subquery decomposition rather than derived from a single LLM-expanded query; and the optimization target is information coverage rather than general relevance. No trained SoftQE checkpoint is publicly available. The two systems also start from different base encoders: SoftQE’s reported BEIR average is substantially below SPLADE-v3’s.

6 Conclusion

We proposed SPLADE-COVER, a learned sparse retrieval model whose expansion mechanism is trained to be information coverage-aware. SPLADE-COVER is trained with a distillation objective that drives the query representation to cover the vocabulary support of multiple aspects of a complex query, and produces a coverage-aware representation with no language-model invocation or post-hoc reranking at query time.

In an evaluation on three datasets with nugget-level judgments, SPLADE-COVER often improves over SPLADE-v3 on both coverage-aware metrics and relevance metrics. Reranking-based methods (MMR, SMMR) collapse toward the SPLADE-v3 baseline with gridsearch, meaning that they are not able to improve coverage over the baseline. Generation-based methods are more effective, but they require one or more LLM calls per query plus additional encoder passes for the expanded query material, at roughly an order of magnitude more query-time latency. The substantial gap between Subquery Oracle and all non-Oracle methods suggests that aspect quality may be a major bottleneck for first-stage coverage retrieval, motivating further work on training-time objectives that improve aspect representations in the learned retriever.

Limitations

Due to the limited number of nugget or aspect-level retrieval benchmarks, our evaluation is limited

to three datasets, two of which have small query counts that limit the statistical power of comparisons (NeuCLIR24, 19 queries; TREC Interactive, 20 queries).

SPLADE-COVER is currently a monolingual (English) model: it is initialized from English SPLADE-v3 and trained on English subqueries, and the cross-lingual NeuCLIR24 collection is evaluated through its English query and nugget translations. Our training supervision is generated by a single large language model (Llama 3.3 70B), and the resulting synthetic complex queries may not fully represent the complete distribution of user information needs. Finally, all reported results use a single training seed; we do not characterize variance across seeds.

References

- Akari Asai, Zequi Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.
- Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. 2024. Efficient inverted indexes for approximate retrieval over learned sparse representations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 152–162.
- Jaime Carbonell and Jade Goldstein. 1998. 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.
- Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.
- Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. [Novelty and diversity in information retrieval evaluation](#). In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–666.
- Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’09)*, pages 758–759.

- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020)*. ArXiv:2102.07662.
- Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference (TREC 2019)*. ArXiv:2003.07820.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kaustubh D. Dhole and Eugene Agichtein. 2024. [Gen-QREnsemble: Zero-Shot LLM Ensemble Prompting for Generative Query Reformulation](#), pages 326–335. Springer Nature Switzerland.
- Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural IR models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.
- William R. Hersh and Paul Over. 2000. TREC-8 interactive track report. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, NIST Special Publication 500-246. National Institute of Standards and Technology.
- Jia-Huei Ju, François G. Landry, Eugene Yang, Suzan Verberne, and Andrew Yates. 2026. [LANCER: LLM reranking for nugget coverage](#). In *Advances in Information Retrieval (ECIR 2026)*, volume 16484 of *Lecture Notes in Computer Science*. Springer.
- Jia-Huei Ju, Suzan Verberne, Maarten de Rijke, and Andrew Yates. 2025. Controlled retrieval-augmented context evaluation for long-form RAG. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.
- Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. SPLADE-v3: New baselines for SPLADE. *arXiv preprint arXiv:2403.06789*.
- Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldaini, and Eugene Yang. 2025. Overview of the TREC 2024 NeuCLIR track. *arXiv preprint arXiv:2509.14355*.
- Yibin Lei, Tao Shen, and Andrew Yates. 2025. ThinkQE: Query expansion via an evolving thinking process. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.
- Kiryl Liakhnovich, Oleg Lashinin, Andrei Babkin, Michael Pechatov, and Marina Ananyeva. 2025. [SMMR: Sampling-based MMR reranking for faster, more diverse, and balanced recommendations and retrieval](#). In *Proceedings of the 19th ACM Conference on Recommender Systems (RecSys '25)*.
- Antonio Mallia, Torsten Suel, and Nicola Tonellotto. 2024. Faster learned sparse retrieval with block-max pruning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2411–2415.
- Thong Nguyen, Yibin Lei, Dylan JH Ju, Eugene Yang, and Andrew Yates. 2026. MILCO: Learned sparse retrieval across languages via a multilingual connector. In *International Conference on Learning Representations*, volume 2026, pages 46623–46647.
- Thong Nguyen, Sean MacAvaney, and Andrew Yates. 2023. A unified framework for learned sparse retrieval. In *Advances in Information Retrieval (ECIR 2023)*.
- Paul Over. 1998. The TREC-6 interactive track report. In *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*, NIST Special Publication 500-240. National Institute of Standards and Technology.
- Paul Over. 1999. TREC-7 interactive track report. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, NIST Special Publication 500-242. National Institute of Standards and Technology.
- Varad Pimpalkhute, John Heyer, Xusen Yin, and Sameer Gupta. 2024. SoftQE: Learned representations of queries expanded by LLMs. In *Advances in Information Retrieval (ECIR 2024)*. ArXiv:2402.12663.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Corby Rosset, Ho-Lam Chung, Guanghui Qin, Ethan C. Chau, Zhuo Feng, Ahmed Awadallah, Jennifer Neville, and Nikhil Rao. 2025. [Researchy questions: A dataset of multi-perspective, compositional questions for deep research](#). In *Proceedings of the ACM Web Conference 2025*.
- Saron Samuel, Andrew Yates, Dawn Lawrie, Ian Soboroff, Trevor Adriaanse, Benjamin Van Durme, and Eugene Yang. 2026. [CoverageBench: Evaluating information coverage across tasks and domains](#). Preprint, arXiv:2603.20034.

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, pages 881–890.

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2015. Search result diversification. *Foundations and Trends in Information Retrieval*, 9(1):1–90.

Wonduk Seo, Hyunjin An, and Seunghyun Lee. 2025. [A new query expansion approach via agent-mediated dialogic inquiry](#). *Preprint*, arXiv:2502.08557.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Liang Wang, Nan Yang, and Furu Wei. 2023. [Query2doc: Query expansion with large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality*, 10(4):1–20.

Chengxiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–17.

Yunfei Zhong, Jun Yang, Yixing Fan, Lixin Su, Maarten de Rijke, Ruqing Zhang, and Xueqi Cheng. 2025. Reason to retrieve: Enhancing query understanding through decomposition and interpretation. *arXiv preprint arXiv:2509.06544*.

A Generation-Based Baseline Implementation Details

All four generation-based baselines use Qwen3.5-9B (Qwen Team, 2026) served via vLLM with thinking traces disabled. All four baselines share the Qwen3-specific sampling settings $\text{top}_p=0.8$, $\text{top}_k=20$, and $\text{presence_penalty}=1.5$; the remaining parameters are listed per-baseline below.

GenQREnsemble. 10 prompt variants per query (temperature 0.92, max_tokens 256). The keyword outputs from all 10 variants are merged and concatenated with the original query before encoding.

Statistic	Value
Header blocks per row (avg / median / max)	2.07 / 2 / 8
Synthetic report request length (source text; avg / median, tokens)	183.0 / 180
Concatenated header(s) length (target text; avg / median, tokens)	120.5 / 114
Document length (avg / median, tokens)	507.1 / 512
Triples with truncated header/target text (512 tokens)	19 (0.01%)
Triples with truncated document (512 tokens)	248,074 (96.7%)

Table 4: Statistics for the curated training dataset (rating threshold 5, min_matches 1, max_samples 3). 512 tokens is the maximum sequence length fed to SPLADE for each field; sequences longer than that are truncated. All text (report requests, headers, and relevance ratings) is generated by Llama-3.3-70B-Instruct.

AMD-QE. 3 LLM calls per query: Socratic Questioning Agent, Dialogic Answering Agent, and Reflective Feedback Agent (temperature 0.8). The final query concatenates the original (repeated three times) with the refined pseudo-answers.

HyDE. 8 pseudo-documents per query (temperature 0.7, max_tokens 300). Each query uses one of four task-specific bare-bones templates following Gao et al. (2023): NeuCLIR24 and CRUX-DUC04 use the *news* template (Please write a news passage about the topic. Topic: {query} Passage:); TREC Interactive uses the *qa* template (Please write a passage to answer the question. Question: {query} Passage:).

Subquery RRF. 10 subqueries per complex query generated via the prompt in Figure 2. The 10 resulting per-subquery SPLADE-v3 rankings are fused via reciprocal rank fusion with the main query.