MULTIHIT PRINCIPLE FOR LEARNED SPARSE RETRIEVAL: DETERMINISTIC BOUNDS ON CANDIDATE VOLUME AND TOP-K SURVIVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned sparse retrieval such as SPLADE supports semantic search while retaining the efficiency of inverted indexes, but semantic expansion can make a query touch a large fraction of the corpus. We study whether the overlap structure across posting lists can be used as a filter to remove background documents before document scoring. We identify a MultiHit separation: as the required number of matched coordinates increases, the candidate set shrinks approximately exponentially while many highly ranked documents survive. This motivates an a-of-B filter that retains documents matching at least a of the B highest-weight query coordinates. Since SPLADE rankings are determined by weighted inner products rather than hit counts, however, it is nontrivial whether top-ranked documents still survive the filter. We characterize both sides of this separation using finite-corpus statistics. Intersection counts bound the candidate-set size without any distributional assumptions, while score-margin moments give lower bounds on overall Top-K survival. We further formulate threshold selection as finite-sample risk control for future queries from the same distribution. Our end-to-end implementation on MS MARCO, DBpedia, and FiQA achieves 1.47–$3.14\times$ speedups over Naive, with substantially larger speedups in single-query latency over PISA, BMP, and Seismic when using 16 threads, while maintaining over 99% mean Top-1000 survival.

1 INTRODUCTION

Inverted indexes are effective because a query usually touches only a small part of the corpus. However, learned sparse retrieval models such as SPLADE (Formal et al., 2021b;a; Lassance et al., 2024) use neural expansion to activate many semantically related coordinates that may not occur in the document, leading to much longer posting lists than traditional methods such as BM25 (Robertson & Zaragoza, 2009). More importantly, there's no retrieval rule that can directly filter low-relevance documents before scoring and comparison, so retriever may need to process a large fraction of the posting-list union.

Most existing methods address this cost through score reasoning, including estimating score upper bounds and perform skipping, like WAND (Broder et al., 2003), MaxScore, and Block-Max WAND (Ding & Suel, 2011), as well as index and query pruning (Lassance & Clinchant, 2022; Lassance et al., 2023) and blockwise summaries (Bruch et al., 2024b). These methods share a common principle: candidate work is reduced by estimating whether additional score computation can still affect the final ranking. We ask a more basic question: *is there a useful filtering signal in learned sparse representations before any score computation?*

Our starting observation is that there is a clear separation in hit counts between Top-K documents and background documents. For example, for SPLADE-v3 on MS MARCO at $K = 100$, increasing the threshold from 3-of-8 to 4-of-8 reduces the candidate set to about 30% of its previous size while retaining 99.5% of the current Top-100 documents. We call this phenomenon the *MultiHit principle*. This motivates the *a-of-B filter*: select the B highest-weight query coordinates, retain documents appearing in at least a of their posting lists.

The effectiveness of this rule is not obvious. SPLADE ranks documents by weighted inner products, whereas MultiHit uses only binary support counts; a high-scoring document may concentrate most of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: MultiHit separation at fixed $B = 10$ using SPLADE-v3. All entries are average percentages retained (%). Bold entries mark the largest displayed a preserving at least 99% of Top-10.

	MS MARCO					DBPedia-Entity					FiQA				
a	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Corpus	30.03	10.82	4.11	1.57	**0.58**	39.04	14.99	6.10	**2.59**	1.05	72.26	43.83	23.51	**11.40**	5.00
Top-10	100.00	100.00	99.97	99.87	**99.36**	100.00	100.00	100.00	**99.40**	96.98	100.00	100.00	99.80	**99.01**	95.99
Top-100	100.00	100.00	99.82	98.83	95.67	100.00	100.00	99.68	97.00	91.69	100.00	99.92	99.21	95.80	85.96
Top-1000	100.00	99.84	98.48	94.61	85.43	100.00	99.64	96.98	89.46	77.02	100.00	99.13	94.75	82.77	62.00

its score on only a few coordinates. Moreover, posting lists are strongly dependent, so rapid candidate reduction cannot be derived directly from independence or sparsity. This raises two questions.

Q1: Why does the candidate background shrink? Can we quantify this reduction directly from corpus overlap statistics, without making distributional assumptions about the posting lists?

Q2: Why do top-ranked documents survive? Can we explain their survival from the score contributions of the selected coordinates, rather than from observed survival rates alone?

We address the two questions separately. For Q1, we formulate candidate volume as a discrete moment problem and derive sharp deterministic bounds from low-order intersection counts. We further extend these bounds to sampled statistics with confidence corrections. For Q2, we use score-margin analysis to separate survival supported by substantial score contributions from survival that depends on weak additional hits, and derive moment bounds.

Q1 and Q2 meet at the selection of threshold a: can we choose a from a small sample set while explicitly controlling ranking loss? We use separate criteria for mean loss and bad-query probability.

Finally, once a feasible threshold can be selected for each corpus, we ask whether its change with corpus size and ranking depth is predictable. Across nested corpora, a follows an integer staircase that is well summarized by $a_{N,K} \simeq c + s \log(N/K)$. This finite-range model predicts larger-corpus operating points from smaller corpora reasonably well.

2 The MultiHit Principle

MultiHit filtering affects the corpus background and high-ranked documents differently as the hit requirement grows, since most corpus documents match few selected query coordinates, while high-ranked documents tend to match more of them.

Table 1 shows this effect for SPLADE-v3 across three datasets and three ranking depths, strongest on MS MARCO, weaker on DBPedia-Entity, and weakest on FiQA. It also changes with ranking depth, so there is no single choice of B and a that works best everywhere. This leads to three questions: (i) how quickly does the candidate set shrink; (ii) does high-ranked documents always survive; and (iii) how to choose the filter threshold a?

Setup Let $\mathcal{D} = \{d_1, \ldots, d_N\}$ be a corpus of N documents. SPLADE Formal et al. (2021b) represents each document d and query q by a nonnegative sparse vector $\mathbf{x}_d, \mathbf{x}_q \in \mathbb{R}^{|\mathcal{V}|}_{\geq 0}$ over a fixed vocabulary $\mathcal{V}$. Documents are ranked by the inner product $s(q, d) = \langle \mathbf{x}_q, \mathbf{x}_d \rangle$. For each vocabulary word w, the inverted index stores the posting list $I_w = \{d \in \mathcal{D} : x_{d,w} > 0\}$.

Let $W_B(q) = (w_1, \ldots, w_B)$ be the B largest positive coordinates of x_q. The number of matched query keywords in document d is called "hits", $H_B(q, d)$. The a-of-B filter keeps documents with at least a hits: $C_a(q) = \{d \in \mathcal{D} : H_B(q, d) \geq a\}$ where documents are then scored and ranked. The way that filter determines whether a document should be scored is independent of other documents.

2.1 Q1: How quickly does candidate set shrink

Table 1 shows that as hits r increases, candidate population C_r decrease at a rate close to exponential decay. Although this usually happens when selected posting lists are independent, both the semantic relations among SPLADE terms and statistical experiments show that these posting lists are highly

Table 2: Scalar-2 bounds at $B = 6$. Full columns use exact intersection totals. For sampled columns, L_r^- and U_r^+ include a 95% confidence margin, so that $L_r^- \le L_r$ and $U_r^+ \ge U_r$. Results use 8,192 or 32,768 posting-appearance draws and report medians over 20 independent repetitions. Counts refer to means over the fixed query set. The 95% guarantee holds within each repetition.

			Candidate-count bounds					Contraction bounds		
Dataset	r	C_r	L_r/C_r Full	U_r/C_r Full	U_r^+/C_r 8K	U_r^+/C_r 32K	C_{r+1}/C_r	U_{r+1}/L_r Full	U_{r+1}^+/L_r^- 8K	U_{r+1}^+/L_r^- 32K
MS MARCO	1	1726.93	0.92	1.18	1.19	1.18	0.24	0.35	0.38	**0.37**
	2	423.08	0.82	1.32	1.40	1.36	0.22	0.36	0.49	**0.41**
	3	92.77	0.78	1.34	1.55	1.44	0.19	0.29	0.74	**0.43**
	4	17.68	0.82	1.19	1.66	1.39	0.13	0.16	1.00	**0.35**
DBpedia-Entity	2	293.66	0.79	1.35	1.43	1.39	0.24	0.42	0.55	**0.47**
	3	69.63	0.79	1.39	1.57	1.47	0.21	0.32	0.74	**0.42**
FiQA	2	12.23	0.64	1.49	1.55	1.52	0.33	0.76	0.87	**0.81**
	3	3.99	0.55	1.49	1.63	1.55	0.26	0.63	0.82	**0.71**

dependent. It still remains to know how much of this decrease can be proved from only low-order information without any assumption on data distribution.

Fix a query and its B selected posting lists $I_1, \dots, I_B$. Let n_h denote the number of documents matching exactly h keywords and the number of documents with hits no less than r, $C_r := |C_r(q)|$ can also be $C_r = \sum_{h=r}^{B} n_h$. Size of all j-way intersections is $M_j = \sum_{h=j}^{B} \binom{h}{j} n_h$ since M_r repeatedly counts a document $\binom{h}{r}$ times. Applying the classical results for finite-support discrete moment problems (Prékopa, 1990), we obtain the tightest possible bounds on C_r using binomial moments M_r/N and M_{r+1}/N (we call M_r and M_{r+1} two totals), without any distributional assumptions.

Theorem 1 (Scalar-2: Optimal bounds from binomial moments (Prékopa, 1990)). *For $1 \le r < B$, $M_r > 0$, let $\mu = (r+1)M_{r+1}/M_r$. Let $k = \lfloor \mu \rfloor$, $\psi_r(\mu) = \frac{k+1-\mu}{\binom{r+k}{r}} + \frac{\mu-k}{\binom{r+k+1}{r}}$. Then*

$$L := M_r \psi_r(\mu) \;\le\; C_r \;\le\; M_r \left[1 - \frac{\mu}{B-r} \left(1 - \frac{1}{\binom{B}{r}} \right) \right] =: U.$$

Proof sketch. Define $p_k = \binom{r+k}{r} n_{r+k}/M_r$ for $0 \le k \le B-r$, satisfying $\sum_k p_k = 1$ and $\sum_k k p_k = \mu$, while $C_r/M_r = \sum_k p_k / \binom{r+k}{r}$. The two bounds are the solutions of two linear programs over $\mathcal{P}_\mu = \{p \ge 0 : \sum_k p_k = 1,\ \sum_k k p_k = \mu\}$: $L/M_r = \min_{p \in \mathcal{P}_\mu} \sum_k p_k \psi_r(k) = \psi_r(\mu)$ and $U/M_r = \max_{p \in \mathcal{P}_\mu} \sum_k p_k \psi_r(k) = 1 - \frac{\mu}{B-r}(1 - \binom{B}{r}^{-1})$. By convexity of ψ_r, the two optimal bounds are attained with $\mathrm{supp}(p_{\min}) \subseteq \{\lfloor \mu \rfloor, \lceil \mu \rceil\}$ and $\mathrm{supp}(p_{\max}) \subseteq \{0, B-r\}$, respectively. Reconstructing $n_{r+k} = M_r p_k / \binom{r+k}{r}$ preserves both given totals, establishing sharpness in the nonnegative real-valued relaxation.

Estimating contraction from sampled posting appearances. For a fixed set of queries, we sample uniformly with replacement from all posting appearances in the selected lists. If a query–document pair has h selected hits, it appears h times in this sampling space. We therefore use a factor of $1/h$ to correct for this repeated appearance when estimating intersection totals.

We do not treat the estimated values themselves as valid bounds. Instead, we build one-sided empirical Bernstein confidence bounds for the linear functions used by the Scalar-2 upper bound and by each segment of the lower bound. In each repetition, we split the total failure probability $\delta = 0.05$ across all tested orders, lower-bound segments, and sampling budgets. This gives simultaneous bounds $L_r \le \bar{C}_r \le U_r$, where $\bar{C}_r$ is the mean candidate count over the fixed set of queries.

Table 2 shows that after 32,768 random samples, estimated upper bound remains $1.18\times$–$1.55\times$ true mean candidate count. Moreover, all $U_{r+1}/L_r < 1$. For example, on MS MARCO Dev-small, sampled U_{r+1}/L_r for the four adjacent steps are approximately $0.37, 0.41, 0.43, 0.35$ and the candidate count is less than $0.43^3 \approx 0.08$ of its original size. The same trend is also observed on DBpedia-Entity and FiQA.

Table 3: Two-moment analysis of MultiHit survival on held-out queries. All entries are percentages.

Dataset	(K, B, a)	$\tau = 0$ (survival) Surv.	LB	$\tau = 1\%$ F_τ	UB	Weak	$\tau = 5\%$ F_τ	UB	Weak	$\text{Tail}_{1\%}$
MS MARCO	$(10, 10, 5)$	98.90	94.15	1.38	6.87	0.29	4.43	13.49	3.33	3.91
	$(100, 8, 3)$	99.23	93.57	0.83	6.94	0.06	1.51	9.63	0.75	11.52
	$(1000, 8, 2)$	99.77	95.23	0.25	5.01	0.02	0.44	6.16	0.21	16.30
	$(1000, 10, 2)$	99.89	96.12	0.12	4.06	0.01	0.22	4.91	0.11	10.12
DBpedia-Entity	$(10, 10, 5)$	96.57	89.61	4.00	12.41	0.57	8.54	22.06	5.11	17.10
	$(100, 8, 3)$	98.95	92.83	1.14	7.83	0.09	2.08	11.34	1.03	25.18
	$(1000, 8, 2)$	99.26	94.45	0.78	5.86	0.04	1.02	7.35	0.29	31.42
	$(1000, 10, 2)$	99.51	95.73	0.51	4.47	0.02	0.61	5.41	0.12	16.44
FiQA	$(10, 10, 5)$	95.57	84.83	5.19	17.91	0.76	11.95	30.97	7.52	26.42
	$(100, 8, 3)$	96.36	86.40	4.02	15.04	0.38	6.79	22.39	3.15	44.75
	$(1000, 8, 2)$	97.27	88.16	2.93	12.68	0.20	4.28	16.82	1.55	59.68
	$(1000, 10, 2)$	99.19	92.97	0.89	7.43	0.07	1.39	9.38	0.58	53.83

Notes. **Surv.**: actual exhaustive Top-K survival. **LB**: two-moment lower bound $1 - \Phi_2(0)$. F_τ: fraction of Top-K documents with $X \le \tau$. **UB**: sharp two-moment upper bound $\Phi_2(\tau)$. **Weak**: $F_\tau - F_0$, the fraction that survives the filter but whose contributions beyond the largest $a-1$ selected contributions sum to at most τ of the full score. **Tail@1%**: query-tail score fraction among weak-margin survivors $0 < X \le 1\%$, averaged over queries containing at least one such document.

2.2 Q2: Can Highly Ranked Documents Survive MultiHit Filtering?

Q1 explains why increasing hits can rapidly shrink the candidate set, but it does not use the SPLADE ranking score and high-ranked documents may be removed at the same rate. We aim to prove MultiHit filter will save as many documents as using Full SPLADE vectors. We find the score structure of exhaustive Top-K documents itself implies the survival.

For an exhaustive Top-K document d of query q, let $z_1 \ge \cdots \ge z_B$ be its sorted score contributions on B highest-weighted query coordinates that $z_j = x_{q,w_j} x_{d,w_j} \ge 0$, and define MultiHit score margin $X_{B,a}(q,d) := \frac{\sum_{i=a}^{B} z_i}{s(q,d)} \in [0,1]$., we have $X_{B,a}(q,d) > 0 \iff H_B(q,d) \ge a$. MultiHit survival problem can be written directly as a probability about score mass. For any $d \in \text{TopK}(q)$, let $X = X_{B,a}(q,d)$ and we have $R_{B,a,K}(q) = \Pr(X > 0) = \frac{|\text{TopK}(q) \cap C_a(q)|}{K}$.

More generally, define $F_\tau(q) := \Pr(X \le \tau)$ and $R_{B,a,K}(q) = 1 - F_0(q)$. Let $\mu_q := \mathbb{E}[X]$ and $\nu_q := \mathbb{E}[X^2]$. We now question: among all probability distributions on $[0,1]$ with these two moments, how large can F_τ be? This is a bounded univariate moment problem (Bertsimas & Popescu, 2005).

Theorem 2 (Two-moment sharp bound (Bertsimas & Popescu, 2005)). *For $0 < \mu < 1$, $\mu^2 \le \nu \le \mu$,*

$$F_\tau(q) \le \Phi_2(\tau; \mu, \nu) := \begin{cases} \dfrac{\nu - \mu^2}{\nu - 2\mu\tau + \tau^2}, & 0 \le \tau < (\mu - \nu)/(1 - \mu), \\ 1 - \dfrac{\nu - \mu\tau}{1 - \tau}, & (\mu - \nu)/(1 - \mu) \le \tau < \nu/\mu, \\ 1, & \tau \ge \nu/\mu. \end{cases}$$

At $\tau = 0$, Theorem 2 reduces to $F_0(q) \le 1 - \frac{\mu_q^2}{\nu_q}$, and $R_{B,a,K}(q) \ge \frac{\mu_q^2}{\nu_q}$, the classical second-moment (Paley–Zygmund) bound (Paley & Zygmund, 1932). Hence, the first two moments of the score margin alone impose a deterministic lower bound on aggregate Top-K survival.

Table 3 shows the two-moment theorem provides a conservative but nontrival distribution-free lower bound on Top-K survival. Across all 12 settings, the observed survival ranges from 95.57% to 99.89%, while the first two moments of the score margin X alone explains 84.83%–96.12%. Besides counting how many documents survive the filter, the margin analysis shows whether survival is supported by large score contributions or only by several weak extra hits, which gives an offline way to compare different filtering settings.

However, $X > 0$ alone does not tell us whether the a-th hit is necessary, since its score contribution z_a can be numerically tiny. We next test $\tau > 0$, since $X \le \tau$ means that all selected contributions

Table 4: Joint (B, a) search and risk-controlled sample for SPLADE-v3. For the joint search, each cell reports a / candidate fraction (%), where a is the largest threshold attaining at least 99% mean Top-K survival. For finite-sample, $B = 10$ and $n = 100$ queries; CRC targets expected miss fraction $\leq 5\%$, while LTT-FS targets $\Pr(\text{survival} < 95\%) \leq 5\%$ with 95% calibration confidence.

		Joint search: a / candidate (%)							Risk-controlled sample ($B = 10$)			
Dataset	K	$B = 4$	$B = 6$	$B = 8$	$B = 10$	$B = 12$	$B = 18$	$B = 20$	Method	a (count)	Cand.	Bad
MS MARCO	10	2/1.96	3/1.05	4/0.75	**5/0.58**	5/1.10	6/1.60	6/1.99	CRC	6 (18/20)	0.24	10.11
									LTT-FS	4 (18/20)	2.20	0.81
	100	1/12.78	2/4.78	**3/2.47**	3/4.11	4/2.52	4/5.37	4/6.20	CRC	4 (17/20)	1.45	9.08
									LTT-FS	3 (13/20)	6.55	0.82
	1000	1/12.78	1/19.53	**2/7.87**	2/10.82	2/13.44	3/10.05	3/11.21	CRC	3 (20/20)	4.17	9.78
									LTT-FS	2 (19/20)	11.94	0.50
DBpedia	10	1/13.36	2/6.33	3/3.66	**4/2.59**	4/4.05	5/5.06	5/6.00	CRC	5 (19/20)	1.12	10.53
									LTT-FS	3 (19/20)	5.91	0.23
	100	1/13.36	2/6.33	2/10.46	**3/6.10**	3/8.62	4/8.99	4/10.30	CRC	4 (18/20)	2.94	10.47
									LTT-FS	2 (13/20)	11.88	0.87
	1000	1/13.36	1/23.12	**2/10.46**	2/14.99	2/19.31	2/30.68	2/32.95	CRC	3 (18/20)	6.98	12.22
									LTT-FS	2 (10/20)	27.07	0.88
FiQA	10	1/38.67	2/21.23	3/14.50	**4/11.40**	4/18.55	6/17.86	7/15.05	CRC	5 (12/20)	7.55	11.73
									LTT-FS	3 (18/20)	23.87	0.80
	100	1/38.67	1/52.25	2/32.50	**3/23.51**	3/32.87	5/27.09	5/32.28	CRC	4 (12/20)	16.22	11.47
									LTT-FS	2 (16/20)	39.70	0.87
	1000	**1/38.67**	1/52.25	1/63.25	2/43.83	2/53.50	3/54.35	3/59.09	CRC	2 (17/20)	40.71	6.32
									LTT-FS	1 (18/20)	69.38	0.43

Notes. For finite-sample, a gives the modal selected threshold and its frequency over 20 splits; both modes are reported when tied. Cand. and Bad are mean held-out candidate and bad-query (%). All calibration splits select a threshold.

beyond the largest $a - 1$ together account for at most a τ fraction of the full document score and $F_\tau - F_0 = \Pr(0 < X \leq \tau)$ measures documents that survive MultiHit but have only a small score margin from their additional hits.

From Table 3, such documents are rare. At $\tau = 1\%$, the weak-margin survivor fraction is only 0.01%–0.76% across all 12 settings. At $(10, 10, 5)$, only 0.29%, 0.57%, and 0.76% on three datasets of Top-10 documents satisfy $0 < X \leq 1\%$. So a-th or more hits contribute a non-negligible amount of score. Even at the looser threshold $\tau = 5\%$, the weak-margin fraction remains only 0.11%–7.52%, showing that this conclusion is not specific to the 1% threshold.

The two-moment bounds still leave a gap to the observed survival. At $(10, 10, 5)$, for example, the actual miss fractions are 1.10%, 3.43%, and 4.43%, whereas the corresponding two-moment upper bounds are 5.85%, 10.40%, and 15.17%. To understand this, we inspect the documents closest to the survival boundary, namely those with $0 < X \leq 1\%$, few of which are in $\text{TopK}(q)$, and on FiQA their scores rely more strongly on query coordinates outside Top-B. At $(10, 10, 5)$, the query-tail score fraction among these documents increases from 3.91% on MS MARCO to 17.10% on DBpedia-Entity and 26.42% on FiQA. Although two moments are insufficient to fully characterize the score-margin distribution, high-ranked documents that survive only through a numerically tiny additional hit are rare.

2.3 Q3: How to Choose Hit Threshold a?

We now study how to choose Hit Threshold a from a finite corpus. We first jointly tune (B, a) under target survival to see B's influence to a. We then fix B and formulate choosing a as a finite-sample risk-control problem, recovering the usual mean-survival criterion, CRC controls expected miss risk, and LTT further controls the fraction of poorly served queries. Finally, we measure how much calibration data and candidate-set overhead these stronger guarantees require.

Choosing B first. A larger B can support a larger a, but this does not always imply stronger filter. Table 4 (Left) reports the largest a attaining 99% mean Top-K survival for each B and its candidate fraction. For Top-10, increasing B from 10 to 20 raises acceptable a from 5 to 6 on MS MARCO, but the candidate fraction increases from 0.58% to 1.99%. The same pattern appears on DBpedia-Entity

($a: 4 \to 5$, $2.59\% \to 6.00\%$) and FiQA ($a: 4 \to 7$, $11.40\% \to 15.05\%$). a/B also varies with dataset and retrieval depth and a fixed a/B ratio does not define a common operating point. So we choose B directly from the accuracy–latency trade-off of Top-B query truncation and select a after B is fixed.

Choose a from finite queries. Fix (B, K), define the miss loss of query q as $L_a(q) := 1 - \frac{|\mathrm{TopK}(q) \cap C_a(q)|}{K} \in [0, 1]$. Since $C_{a+1}(q) \subseteq C_a(q)$, $L_a(q)$ is nondecreasing (monotone) in a. Sample n queries, select $\max\{a : n^{-1} \sum_{i=1}^{n} L_a(q_i) \leq \epsilon\}$. To provide finite-sample guarantee for future queries, we first add Conformal Risk Control (CRC) (Angelopoulos et al., 2024) to control the marginal expected miss loss. CRC accepts a when $\frac{\sum_{i=1}^{n} L_a(q_i)+1}{n+1} \leq \epsilon$.

Controlling mean risk alone is not sufficient, since retrieval quality can still be poor for some tail queries. Learn-then-Test (LTT) (Angelopoulos et al., 2025) helps by running hypothesis test for each a: a is certified only when bad-query probability in the sampled data is lower than target.

Specifically, define query's "bad" signal $Z_a(q) = 1$ if its Top-K miss loss $L_a(q) > \gamma$, and 0 otherwise. Let Q denote a future query drawn from the same query distribution, and let $p_a := \Pr[L_a(Q) > \gamma] = \mathbb{E}[Z_a(Q)]$ be its bad-query probability.

We require $p_a \leq \alpha = 0.05$ with confidence $1 - \delta = 0.95$. Sample n queries, let $m_a = \sum_{i=1}^{n} Z_a(q_i)$ be the observed number of bad queries. We test whether $p_a > \alpha$ using the exact binomial lower-tail probability $P_a = \Pr_{Y \sim \mathrm{Binomial}(n,\alpha)}[Y \leq m_a]$, and certify a when $P_a \leq \delta$. Since p_a is monotone, we test sequence $a = 1, \ldots, B$ at level δ and stop at the first a that cannot pass the tests. This fixed-sequence test avoids a Bonferroni penalty while retaining the finite-sample guarantee.

Table 4 (Right) reports the finite-sample calibration results with $B = 10$ and $n = 100$. All calibration splits select a threshold under both CRC and LTT-FS, showing that 100 calibration queries are sufficient to obtain nontrivial operating points under these risk targets.

The two rules have different filtering strengths. Under same miss loss, CRC allows larger a and smaller candidate sets but more bad query. For example, on MS MARCO Top-10, CRC most often selects $a = 6$ and retains only 0.24% of the corpus, whereas LTT-FS selects $a = 4$ and retains 2.20%. However, 10.11% of CRC test queries are bad, compared with only 0.81% under LTT-FS.

The cost of tail protection also grows with retrieval depth. For MS MARCO, the LTT-FS decreases from $a = 4$ at Top-10 to $a = 2$ at Top-1000, while the candidate fraction increases from 2.20% to 11.94%. At FiQA Top-1000, LTT-FS even falls back to $a = 1$. Finite-sample tail protection requires more candidates, especially for deeper retrieval and difficult datasets.

2.4 FINITE-RANGE SCALING

In Table 1 and Table 2, we can see candidate C_r decrease at a rate close to exponential decay as hits r increases. Let $a_{N,K}$ be the largest hit threshold that keeps the mean Top-K survival at the target τ, we test whether $a_{N,K} \simeq c + s \log(N/K)$. The use of N/K also has a simple meaning. A larger corpus makes the filtering problem harder, while a larger K asks us to preserve more documents and therefore allows less aggressive filtering. Thus N and K act in opposite directions.

Figure 1 summarizes the finite-range behavior of three sparse encoders on five nested MS MARCO corpora containing 0.5M, 1M, 2M, 4M, and 8.84M documents. We fix $B = 10$ and select the largest integer threshold a achieving at least 99% mean Top-K survival. For visualization, we use the same five corpus sizes for each $K \in \{10, 100, 1000\}$ and jointly fit $a = c + s \ln(N/K)$ to the resulting 15 points by ordinary least squares. Since N/K changes mainly with K, data points spread evenly along the horizontal axis. Across all three encoders, a increases approximately linearly with $\log(N/K)$ over the tested range, with similar fitted slopes of 0.431–0.485.

We also tested whether separate coefficients for $\ln N$ and $\ln K$ improve prediction using the same nested-corpus measurements. In leave-one-corpus-size-out evaluation across the tested (B, τ) settings, this additional flexibility did not improve mean absolute error: errors increased from 0.266, 0.294, and 0.269 to 0.273, 0.305, and 0.276 hits, respectively. This supports the simpler ratio model as an empirical description, rather than implying that ranking depth has little effect.

Predicting larger corpora from smaller ones. We further evaluate forward prediction, excluding all measurements at or above the target corpus size from fitting. With $B = 10$, a 99% survival target,

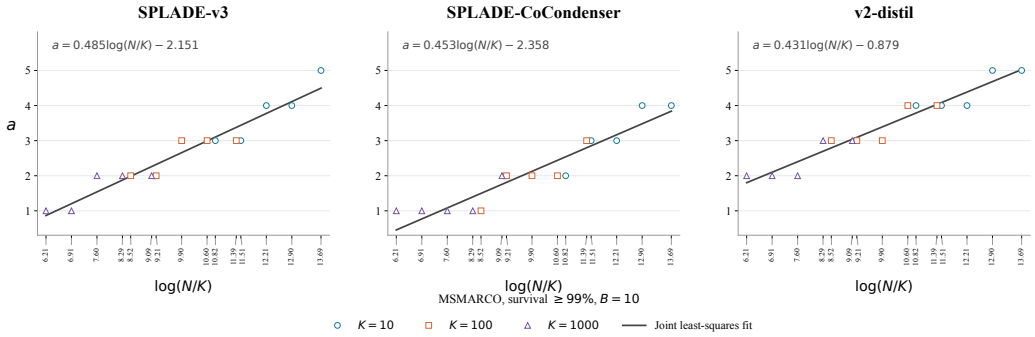

Figure 1: Finite-range test of the logarithmic model on nested MS MARCO corpora with $B = 10$, $K \in \{10, 100, 1000\}$, and target mean survival $\tau = 99\%$. Points show the measured largest feasible threshold $a_{N,K}$. Each line is a joint least-squares fit to all 15 points across the three ranking depths for one encoder, plotted against $\ln(N/K)$.

and $K \in \{10, 30, 100, 300, 1000\}$, we fit the ratio model jointly across ranking depths using only smaller corpora, then predict the threshold at the next size. Predictions are rounded to the nearest integer and clipped to $1 \leq a \leq B$.

The 4 corpus sizes and 5 ranking depths gives 20 out-of-range predictions per encoder. The linear model exactly predicts 18/20 thresholds for SPLADE-v3, 16/20 for CoCondenser, and 20/20 for v2-distil, for a total of 54/60 exact predictions. All six errors are only 1 hit away from the measured threshold, better than a baseline that ignores corpus growth, which predicts the rounded mean threshold for the same K over the available smaller corpora recovers only 26/60..

3 System Application: Low-Latency MultiHit Retrieval

We next study whether the MultiHit rule can be turned into a practical retrieval engine. The main opportunity is that the a-of-B decision depends only on posting membership: whether a document becomes a candidate needs to be decided only once and depends only on that document itself, rather than on other documents. This decision can therefore be made in parallel.

Experimental Setup. All methods retrieve Top-1000 using SPLADE-v3 on the same Intel Xeon Platinum 8163 machine (32 logical CPUs, ∼128 GB RAM). We evaluate on MS MARCO Passage (Bajaj et al., 2018)[1] (8.84M documents), DBpedia-Entity (Hasibi et al., 2017)[2] (4.64M documents), and FiQA (Maia et al., 2018)[3] (57.6K documents), covering web passage retrieval, entity retrieval, and financial question answering. For DBpedia and FiQA, we use the corpora distributed by BEIR (Thakur et al., 2021)[4], with queries and relevance judgments from their `test` splits.

Baselines. We compare MultiHit with several dynamic score-bound pruning methods and Seismic with and without graph expansion. PISA Block-Max WAND (Ding & Suel, 2011; Mallia et al., 2019)[5] uses block-level score upper bounds to skip document ranges that cannot exceed the current top-K threshold while MaxScore (Turtle & Flood, 1995) uses per-coordinate score upper bounds. BMP (Mallia et al., 2024)[6] partitions documents into blocks of size b and sorts the score upper bounds in decreasing order, enabling early termination. Seismic (Bruch et al., 2024b)[7] prunes inverted lists, clusters entries with a heap factor and puncture by clusters before full document scoring.

[1] https://microsoft.github.io/msmarco/
[2] https://public.ukp.informatik.tu-darmstadt.de/thakur/BEIR/datasets/dbpedia-entity.zip
[3] https://public.ukp.informatik.tu-darmstadt.de/thakur/BEIR/datasets/fiqa.zip
[4] https://github.com/beir-cellar/beir
[5] https://github.com/pisa-engine/pisa
[6] https://github.com/pisa-engine/BMP
[7] https://github.com/TusKANNy/seismic

Seismic+κNN (Bruch et al., 2024a) improves by scoring up to `knn` graph neighbors per document. We also use BM25S (Lù, 2024)[8] to construct sparse vectors based on term frequency.

Configurations. For MultiHit, we evaluate fixed operating points $(B, a) = (8, 2)$ on MS MARCO and DBpedia and $(10, 2)$ on FiQA. Both PISA methods use quantized inverted indexes, and BMP uses $b = 8$. For Seismic, length of posting lists is 10,000, the query cut is 8, and the heap factor is 0.8 on MS MARCO and DBpedia, 1 in FiQA. These base search settings were selected using earlier measurements on the same query cohorts, targeting $F@1000 \geq 99\%$, and frozen for the reported experiments. We report `knn=0` as Seismic and `knn=32` as Seismic+κNN. For BM25, Naive and MultiHit ($a = 2$) share the same optimized engine as SPLADE but without Top-B truncation.

Implementation. Our inverted index stores sorted 32-bit document IDs, 16-bit weights, and 64-bit posting offsets in contiguous, memory-mapped arrays. For each query, we scan the Top-B posting lists and increment hit counters per doc; documents with at least a hits become candidates.

We partition sorted document IDs into groups called *tiles*, with approximately four tiles per worker, and schedule them dynamically using OpenMP. Workers reuse counting, scoring, and result buffers. Tiles with no candidates are skipped; docs in the remaining tiles accumulate contributions without checking candidate identity (for speedup) and non-candidates is then filterd. Finally, we merge the per-tile Top-1000 candidates into the exact Top-1000.

3.1 Evaluation

From Table 5, MultiHit shows high retrieval-quality across all evaluated datasets and query sets. We use "T" for the number of threads and "Q" for the number of queries per batch. Compared with the other methods, MultiHit shows no obvious advantage under 1T/Q1 and it has a small advantage under 16T/Q64, around 1.36–$2.57\times$. Its main advantage is observed under 16T/Q1, where parallelization is used within a single query and MultiHit is $8.1\times$–$29.3\times$ faster than the fastest external baseline.

This is because MultiHit is document-level independent: whether one document is filtered depends only on that document, rather than on other documents such as cluster pruning in Seismic and Block Max in BMP. This property enables parallelism inside one query. Naive has the same property, and its 16T/Q1 latency is already lower than several more complex systems.

PISA performs poorly in our experiments, partly because it was designed for traditional BM25 indexes and SPLADE has different weight distribution. Moreover, most methods are much slower than Naive in FiQA even in 1T. With only 57,638 documents, brute force is already cheap, while dynamic pruning pays more for bound checks, candidate management, and result construction. These costs do not shrink with the corpus at the same rate.

Multi-hit uses the same index as Naive, with index sizes of 8.27, 3.90, and 0.079 GiB on MS MARCO, DBPedia-Entity, and FiQA, and nearly identical peak RSS (5.56, 2.56, and 0.28 GiB). Compared with Multi-hit, PISA uses 1.72–1.77$\times$ space and 0.67–1.06$\times$ the peak RSS, Seismic-H uses 2.42–10.49$\times$ index size and 3.74–6.76$\times$ the RSS, and BMP is 2.44–3.08$\times$ large and 3.68–17.72$\times$ the RSS.

Ablation Study. We separate the benefit of filtering from that of execution optimizations through the following ablations. The speedups come from separate comparisons and should not be multiplied.

(1) MultiHit. With the optimized engine fixed at 16T/Q1 on the full evaluation query sets, enabling MultiHit reduces mean latency from 11.790 to 3.753 ms on MS MARCO ($3.14\times$), 7.078 to 2.800 ms on DBPedia ($2.53\times$), and 0.505 to 0.344 ms on FiQA ($1.47\times$). This comparison measures the net benefit of filtering beyond the implementation shared with Naive.

(2) Tile-based execution. On 100 queries per dataset with 16 workers, replacing the original execution layout with tiles reduces mean latency from 11.969 to 6.565 ms on MS MARCO ($1.82\times$), 7.632 to 4.271 ms on DBPedia ($1.79\times$), and 0.655 to 0.412 ms on FiQA ($1.59\times$).

(3) Scoring before filtering. In a separate 100-query-per-dataset, 16-worker experiment holding the tiled engine fixed, removing these checks reduces mean latency from 5.428 to 3.511 ms on MS MARCO ($1.55\times$), 4.112 to 2.601 ms on DBPedia ($1.58\times$), and 0.394 to 0.298 ms on FiQA ($1.32\times$). Checking membership with masks over groups of eight postings gives smaller speedups of $1.17\times$,

[8] https://github.com/xhluca/bm25s

Table 5: Search latency and quality at return depth 1000 using full postings. T/Q denote threads and batch size; latency is amortized ms/query after warmup in Seismic, BMP and PISA.

(a) MS MARCO

	Latency (ms/query)↓			Quality (%)↑			
Method	1T/Q1	16T/Q1	16T/Q64	R@1000	F@1000	M@10	N@10
MS MARCO Dev (6980 queries)							
BM25 (Naive)	116.23	7.50	7.44	85.67	100.00	17.60	22.10
BM25 (MultiHit)	18.34	1.27	1.25	76.76	74.46	17.05	21.25
PISA BMW	467.40	468.58	41.18	97.83	99.49	40.99	47.88
PISA-MS	145.23	145.25	12.07	97.83	99.49	40.99	47.88
Seismic(knn=0)	40.42	66.27	29.39	97.83	99.64	41.08	47.95
Seismic(knn=32)	46.67	61.61	24.23	97.83	99.87	41.08	47.95
BMP ($b = 8$)	37.63	37.55	4.47	97.83	97.04	40.83	47.78
Ours (Naive)	174.37	11.79	11.75	97.83	100.00	41.08	47.95
Ours (MultiHit)	58.34	3.75	3.79	97.83	99.73	41.08	47.95
MS MARCO TREC19 (43 queries)							
BM25 (Naive)	112.18	7.27	7.21	72.75	100.00	79.48	46.46
BM25 (MultiHit)	17.19	1.17	1.18	59.62	70.59	79.48	42.80
PISA BMW	463.07	463.52	39.76	88.06	99.49	85.47	72.18
PISA-MS	145.80	146.08	13.91	88.06	99.49	85.47	72.18
Seismic(knn=0)	38.20	40.69	4.29	87.97	99.52	85.47	72.20
Seismic(knn=32)	47.69	48.05	4.46	87.97	99.78	85.47	72.20
BMP ($b = 8$)	39.22	38.80	7.16	87.36	97.34	85.85	72.71
Ours (Naive)	171.20	14.29	12.04	87.97	100.00	85.47	72.20
Ours (MultiHit)	60.44	4.81	3.92	87.97	100.00†	85.47	72.20
MS MARCO TREC20 (54 queries)							
BM25 (Naive)	117.08	7.59	7.55	73.86	100.00	80.11	47.79
BM25 (MultiHit)	18.67	1.28	1.27	57.83	74.59	79.49	43.24
PISA BMW	503.23	503.90	48.70	90.67	99.45	85.73	75.25
PISA-MS	155.16	155.77	14.05	90.67	99.45	85.73	75.25
Seismic(knn=0)	41.72	42.45	3.61	90.67	99.52	85.73	75.13
Seismic(knn=32)	49.64	50.19	4.12	90.67	99.81	85.73	75.13
BMP ($b = 8$)	39.87	39.88	4.19	90.64	96.80	85.73	75.37
Ours (Naive)	174.39	14.12	12.03	90.67	100.00	85.73	75.13
Ours (MultiHit)	60.14	4.77	3.73	90.67	99.85	85.73	75.13

(b) DBPedia-Entity

	Latency (ms/query)↓			Quality (%)↑	
Method	1T/Q1	16T/Q1	16T/Q64	R@1000	F@1000
DBPedia-Entity (400 queries)					
BM25 (Naive)	52.61	3.93	3.91	66.52	100.00
BM25 (MultiHit)	8.82	0.69	0.68	59.41	67.25
PISA BMW	468.78	470.82	61.63	78.60	99.43
PISA-MS	111.98	113.04	11.74	78.60	99.43
Seismic(knn=0)	40.33	48.73	3.79	78.51	99.40
Seismic(knn=32)	57.09	64.11	4.72	78.52	99.81
BMP ($b = 8$)	68.05	68.13	10.98	78.55	96.75
Ours (Naive)	95.72	7.08	7.04	78.52	100.00
Ours (MultiHit)	39.95	2.80	2.78	78.39	99.39

(c) FiQA

	Latency (ms/query)↓			Quality (%)↑	
Method	1T/Q1	16T/Q1	16T/Q64	R@1000	F@1000
FiQA (648 queries)					
BM25 (Naive)	1.40	0.42	0.41	77.16	100.00
BM25 (MultiHit)	0.84	0.19	0.17	75.36	89.47
PISA BMW	36.47	36.64	3.19	85.87	99.60
PISA-MS	9.93	10.09	0.87	85.87	99.60
Seismic(knn=0)	14.28	77.71	5.53	85.77	99.11
Seismic(knn=32)	33.61	73.52	29.32	85.74	99.55
BMP ($b = 8$)	40.85	41.34	3.95	85.93	98.12
Ours (Naive)	1.96	0.51	0.49	85.87	100.00
Ours (MultiHit)	1.79	0.34	0.34	85.74	99.13

R, F, M, and N denote Recall, fidelity to the corresponding exhaustive retriever, MRR, and nDCG, respectively. All quality values are percentages; nDCG uses linear gains. For SPLADE results, Recall and MRR use relevance ≥ 2 on TREC and ≥ 1 otherwise. For the BM25 TREC19/20 rows, the currently reported Recall and MRR use relevance ≥ 1. M@10 and N@10 are reported only for MS MARCO. PISA-MS denotes MaxScore; Seismic subscripts denote `n_knn`. BMP uses $b = 8$. TREC batch-64 requests contain 43/54 queries. † The unrounded value rounds to 100.00%.

$1.19\times$, and $1.13\times$, respectively, over the same candidate-checked loop. Thus, allowing extra score updates is faster here, although this does not establish that their number is small.

(4) MultiHit also accelerates BM25, but with greater ranking loss. At 16T/Q1, it reduces mean latency from $2.21\times$ to $5.91\times$ in three datasets. However, fidelity to exhaustive BM25 at depth 1000 falls to 74.46%, 67.25%, and 89.47%, while Recall@1000 drops about 8.91%, 7.11%, and 1.80%. There is execution benefit in lexical retrieval, but Learned Sparse Vectors like SPLADE achieve a better balance between candidate filtering and ranking preservation.

4 Related Work

SPLADE learns sparse lexical expansions and impact weights while retaining efficient inverted-index retrieval Formal et al. (2021b;a). However, learned distributions provide fewer skipping opportunities than classical lexical scores Mackenzie et al. (2023), motivating guided traversal Mallia et al. (2022) Mallia et al. (2024), static pruning Lassance et al. (2023), and specialized indexes such as Seismic Bruch et al. (2024b).

Won et al. (2025) also uses Boolean term-threshold filtering but they select the threshold empirically, without theoretical bounds on candidate volume or Top-K survival or formal risk control. Additionally, their evaluation mainly uses the semantic-similarity metric SSS@10 rather than direct Top-K retrieval metrics, and compares only against BM25, without a comprehensive system-level comparison against recent state-of-the-art systems optimized for SPLADE. In fact, MultiHit has long appeared in classical lexical retrieval, for example through Lucene's minimum-should-match interface Apache Software Foundation (2015). However, overlap provides limited ranking preservation: applying the same filter to BM25 substantially reduces Top-1000 fidelity and recall (Table 5) and the behavior of overlap under different distributions induced by learned sparse vectors has not, to our knowledge, been theoretically characterized.

AI USE STATEMENT

We used generative AI tools for literature retrieval and discovery, including locating and checking relevant prior work, as well as for code debugging, translation, and manuscript editing. The core research ideas, proposed methods, and main experimental design were developed by the authors independently. All reported numerical results were obtained from actual experiments.

REPRODUCIBILITY STATEMENT

The source code and scripts required to reproduce the main experimental results are available in an anonymous repository at https://anonymous.4open.science/r/multi-hit-reproduction.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998.

Apache Software Foundation. Apache Lucene 5.2.0: QueryBuilder API. https://lucene.apache.org/core/5_2_0/core/org/apache/lucene/util/QueryBuilder.html, 2015.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv preprint arXiv:1611.09268*, 2018. URL https://arxiv.org/abs/1611.09268v3. Version 3, revised October 31, 2018.

Dimitris Bertsimas and Ioana Popescu. Optimal inequalities in probability theory: A convex optimization approach. *SIAM Journal on Optimization*, 15(3):780–804, 2005. doi: 10.1137/S1052623401399903.

Andrei Z. Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Zien. Efficient query evaluation using a two-level retrieval process. In *Proceedings of the 12th ACM International Conference on Information and Knowledge Management*, pp. 426–434. ACM, 2003. doi: 10.1145/956863.956944.

Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. Pairing clustered inverted indexes with κ-NN graphs for fast approximate retrieval over learned sparse representations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3642–3646. ACM, 2024a. doi: 10.1145/3627673.3679977.

Sebastian Bruch, Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. 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*, pp. 152–162. ACM, 2024b. doi: 10.1145/3626772.3657769.

Shuai Ding and Torsten Suel. Faster top-k document retrieval using block-max indexes. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 993–1002. ACM, 2011. doi: 10.1145/2009916.2010048.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*, 2021a.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 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*, pp. 2288–2292. ACM, 2021b. doi: 10.1145/3404835.3463098.

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. DBpedia-Entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1265–1268. ACM, 2017. doi: 10.1145/3077136.3080751.

Carlos Lassance and Stéphane Clinchant. An efficiency study for SPLADE models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2220–2226. ACM, 2022. doi: 10.1145/3477495.3531833.

Carlos Lassance, Simon Lupart, Hervé Déjean, Stéphane Clinchant, and Nicola Tonellotto. A static pruning study on sparse neural retrievers. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1771–1775. ACM, 2023. doi: 10.1145/3539618.3591941.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. SPLADE-v3: New baselines for SPLADE. *arXiv preprint arXiv:2403.06789*, 2024.

Xing Han Lù. BM25S: Orders of magnitude faster lexical search via eager sparse scoring. *arXiv preprint arXiv:2407.03618*, 2024.

Joel Mackenzie, Andrew Trotman, and Jimmy Lin. Efficient document-at-a-time and score-at-a-time query evaluation for learned sparse representations. *ACM Transactions on Information Systems*, 41 (4):96:1–96:28, 2023. doi: 10.1145/3576922.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. WWW'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of The Web Conference 2018*, pp. 1941–1942. International World Wide Web Conferences Steering Committee, 2018. doi: 10.1145/3184558.3192301.

Antonio Mallia, Michal Siedlaczek, Joel Mackenzie, and Torsten Suel. PISA: Performant indexes and search for academia. In *Proceedings of the Open-Source IR Replicability Challenge (OSIRRC@SIGIR)*, volume 2409 of *CEUR Workshop Proceedings*, pp. 50–56, 2019. URL https://ceur-ws.org/Vol-2409/docker08.pdf.

Antonio Mallia, Joel Mackenzie, Torsten Suel, and Nicola Tonellotto. Faster learned sparse retrieval with guided traversal. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1901–1905, 2022. doi: 10.1145/3477495.3531774.

Antonio Mallia, Torsten Suel, and Nicola Tonellotto. Faster learned sparse retrieval with block-max pruning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2411–2415. ACM, 2024. doi: 10.1145/3626772.3657906.

R. E. A. C. Paley and A. Zygmund. On some series of functions, (3). *Proceedings of the Cambridge Philosophical Society*, 28(2):190–205, 1932. doi: 10.1017/S0305004100010860.

András Prékopa. The discrete moment problem and linear programming. *Discrete Applied Mathematics*, 27(3):235–254, 1990. doi: 10.1016/0166-218X(90)90068-N.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

Howard Turtle and James Flood. Query evaluation: Strategies and optimizations. *Information Processing & Management*, 31(6):831–850, 1995. doi: 10.1016/0306-4573(95)00020-H.

Taeryun Won, Tae Kwan Lee, Hiun Kim, and Hyemin Lee. Efficiency and effectiveness of SPLADE models on billion-scale web document title. *arXiv preprint arXiv:2511.22263*, 2025.