Search Engines

Christian Mølholt Jensen
University of Copenhagen
vdj579@alumni.ku.dk

1 Dataset Analysis

To characterize the corpus, we analyze basic statistics of query and document lengths, measured in word counts using a tokenizer similar to PyTerrier's EnglishTokeniser (Macdonald and Tonellotto, 2020) where we filter out excessively long tokens, digit-heavy terms, and repetitive characters. This removes quotes and apostrophes; manual checks confirmed this does not affect the intent for any of the given queries. We measure lengths in words rather than characters, since this align with IR's term-based indexing and retrieval.

We considered stripping HTML tags and URLs to prevent artificial term inflation. However, this yielded negligible differences and manual inspection revealed very few artifacts; so, for simplicity, we left the raw text intact.

Table 1 summarizes key statistics. Beyond these summary statistics, Figure 1 highlights two key contrasts. First, query lengths are tightly clustered (IQR $\approx$ 4-7 words). Document lengths, in contrast, span several orders of magnitude, with a handful exceeding 300k words. These outliers inflate the mean can disproportionately impact indexing time and retrieval latency.

Statistic	Queries	Documents
Count	4,434	200,000
Min	2	0
Max	22	309,137
Median	5	919
Mean	5.76	1,871.18
Std	2.39	3,498.26

Table 1: Summary statistics of queries and documents.

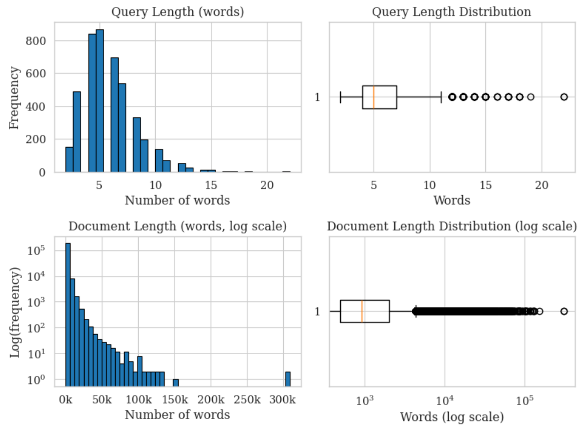

Figure 1: Distribution of query and document lengths.

2 Indexing

2.1 Document Length Filtering

To reduce indexing and latency costs from outlier document lengths, we discard the bottom/top 1% by length ($\approx$2,000 docs below 48 words and $\approx$2,000 above 16,481 words), thereby removing only 0.29%/0.73% of judged-relevant docs. Although this filtering removes a small number of judged-relevant documents, our MSMarco subsample includes only one relevant document per query. As a result, 45 queries (1%) no longer have a relevant document in the filtered collection while we eliminate 2% of documents. We hypothesise that this has negligible impact on retrieval performance. We verify by comparing retrieval performance on the full and filtered collections.

2.2 Experimental Setup

All documents are indexed using PyTerrier (Macdonald and Tonellotto, 2020), an efficient and widely used Python wrapper for the Terrier retrieval system (Macdonald et al., 2012). To reduce variability due to disk I/O and to ensure optimal indexing performance, we preloaded the documents into memory. Each document was formed by concate-

nating its title and body. We build four indexes: (1) full; (2) stopword-removed; (3) stemmed; (4) stopword-removed + stemmed—each with/without length filtering using PyTerriers default parameters. All indexes are built without term position information, as our retrieval scenario does not require phrase or proximity queries.
For each variant, we ran five independent builds to obtain stable build-time statistics. Search time was measured using `process_queries` provided on absalon[1].

All experiments are performed on an Apple Mac M1 (8-core CPU, 7-core GPU) with 16 GB unified memory.

2.3 Indexing Results

Table 2 summarizes the details of the indexing process across the eight index variations.

As indicated by the results the full index retains original document content fidelity but comes with significant drawbacks, such as high storage demands and notably slower build and search times.

Stopword removal produces the largest gains (index $\sim$13% smaller, and $\sim$9× faster search). However, the downside includes potential semantic degradation, as excluding common words might impair contextual comprehension and precision in query results. Stemming yields modest index shrinkage ($\sim$11% smaller), enhancing recall through the consolidation of morphological word variants. However, stemming slightly increases query processing times, likely due to the computational overhead involved. Combining both gives a $\sim$23% smaller index and $\sim$7× faster retrieval. Nonetheless, this combined preprocessing method might amplify semantic loss, potentially diminishing accuracy for certain specialized queries.

Document length filtering reduces index size by $\sim$10% and build time by $\sim$7%. The fastest build ($\sim$187 s) combines filtering, stopword removal, and stemming; the slowest ($\sim$224 s) uses the unfiltered index. Query performance improves notably with the full and stemming-only indexes, while changes with stopword removal are minimal and likely due to randomness.

[1] `process_queries` was altered slightly to handle unseen tokens.

3 Ranking Model Evaluation

3.1 Experimental Setup

We evaluate two ranking models—BM25 and Language Modelleling with Dirichlet smoothing (LMD)—on the each index variant. All experiments use PyTerrier (Macdonald and Tonellotto, 2020). Parameters were tuned using a grid search on the training split of 3,547 queries, and final evaluation was performed on the remaining 887 validation queries. The split was performed using a fixed random seed of 42 to ensure reproducibility.

For BM25, we initially tuned parameters over: $k_1 \in \{1.0, 1.25, 1.5, 1.75, 1.99\}$ and $b \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. These values are based on Büttcher et al. (2010), who suggest $1 \leq k_1 < 2$ (default $k_1 = 1.2$) and $b \in [0, 1]$ (default $b = 0.75$). As performance peaked at $k_1 = 1.99$, we expanded the grid to $k_1 \in \{1.0, 1.25, 1.5, 1.75, 1.99, 2.5, 3.0, 3.5, 4.0, 5.0\}$.

For LMD, we tuned $\mu \in \{500, 1000, 1500, 3000\}$, spanning roughly 0.5–2x the average document length (1667 words). Since $\mu = 500$ consistently performed best, we expanded the grid to $\{50, 100, 250, 500, 1000, 1500, 3000\}$.

Since our dataset comes from MS MARCO, which has sparse binary labels, we optimize for MRR which is also the official metric for the dataset (Craswell et al., 2021).

Metrics used for evaluation include NDCG, MRR, Precision, and Recall at cutoffs 5, 10, and 20, as well as mean response time. To report uncertainty, we compute 95% confidence intervals (CI) as

$$\mathrm{CI} = \frac{s}{\sqrt{n}} t_{1-\alpha/2, n-1}, \quad \alpha = 0.05, \tag{1}$$

where s and n are the sample standard deviation and query count.

We use one-sided paired t-tests to compare each system against the best-performing one (based on mean score) for each metric. To control the family-wise error rate at $\alpha = 0.05$, we adjust the resulting p-values using the Holm-Bonferroni method (Holm, 1979). Systems with adjusted p-values $\geq \alpha$ are considered *not significantly worse* than the best and are highlighted in our tables. Response time is excluded from CI estimation and significance testing due to its heavy-tailed distribution, which can render such statistics unreliable and potentially misleading.

Index Version	Filtered	Docs Indexed	Unique Terms	Total Terms	Avg Doc Length	Size (MB) ↓	Build Time (s) ↓	Search Time (ms) ↓
Full Index	No	200,000	2,912,731	375,471,589	1871.18	607.2	224.4	733.7
Full Index	Yes	195,953	2,629,597	326,621,929	1666.84	529.5	200.1	603.4
Stopwords Removed	No	200,000	2,912,126	222,689,568	1113.45	526.4	206.5	**82.8**
Stopwords Removed	Yes	195,953	2,629,000	194,362,712	991.88	482.0	188.1	103.5
Stemming Only	No	200,000	2,654,799	375,471,589	1871.18	540.9	219.9	853.5
Stemming Only	Yes	195,953	2,404,075	326,621,929	1666.84	472.0	210.5	676.4
Stopwords + Stemming	No	200,000	2,654,647	222,689,568	1113.45	465.6	197.5	104.7
Stopwords + Stemming	Yes	195,953	2,403,925	194,362,712	991.88	**426.8**	**187.3**	103.5

Table 2: Indexing statistics for all four configurations. Bold indicates the best configurations.

3.2 Evaluation Results

Table 3 and 4 present results on the training and validation set, respectively, for the best performing configurations for BM25 and LMD.

The evaluation results demonstrate that preprocessing significantly affects retrieval performance, sometimes more than the choice of retrieval model itself. Across both training and validation, stopword-removed BM25 ($k_1 = 3.5, b = 0.75$) and LMD ($\mu = 100$ with stopwords+stemming) are top–rank leaders. Differences across filtered vs. unfiltered or with-stemming vs. without were generally not significant.

Our optimal BM25 configurations, with k_1 values between 2.5 and 3.5, diverge from the recommended range by Büttcher et al. (2010), but align with other MS MARCO benchmarks, where $k_1 = 3.8$ and $b = 0.87$ performed best (Yang et al., 2017).

For LMD, the optimal smoothing parameter ($\mu = 100$) was much lower than the average document length (∼1,667 words). This can be explained by the skewed distribution of document lengths, where the mode is far below the mean (see Figure 2). Removing stopwords further reduces document lengths, supporting the low optimal μ.

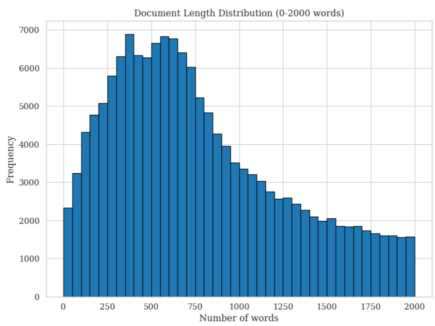

Figure 2: Closer look of the distribution of document lengths.

BM25 notably excels in top-rank focused metrics, highlighting its strength in quickly retrieving relevant documents, essential for datasets with sparse and binary relevance labels like MS MARCO. Its explicit handling of term frequency saturation and document length normalization likely contributes to this advantage over LMD.

LMD, despite trailing BM25 slightly in top-ranked precision, provides competitive performance, especially for recall-oriented metrics. Its smoothing approach might dilute strong relevance signals, particularly in scenarios dominated by sparse, binary judgments.

MRR and nDCG effectively emphasize relevance at higher ranks, aligning closely with user expectations. However, MRR's sensitivity to single-document rankings and nDCG's assumption of graded relevance, which contrasts with MS MARCO's binary labels, limit their interpretability. Precision and Recall provide intuitive measures but are less discriminative at deeper ranks due to data sparsity.

The use of percentile-based filtering for document length notably reduced index size and build time but marginally decreased retrieval effectiveness. The modest trade-off suggests that while excessive filtering risks losing relevant content, moderate filtering can be beneficial depending on the use case.

4 Relevance Feedback & Query Expansion

We start from the best-performing version of BM25 and LMD identified in Section 3 - BM25 ($k_1 = 3.5$, $b = 0.75$) on the index with stopwords removed and without filtering and LMD ($\mu = 100$) on the index with stemming, stopwords removed and without filtering. On top of these two baselines we implement three query–expansion (QE) strategies:

(a) Pseudo relevance feedback using RM3 (Jaleel et al., 2004)

Model (Params)	Index Variant	Filtered	nDCG@5 ↑	nDCG@10 ↑	nDCG@20 ↑	MRR@5 ↑	MRR@10 ↑	MRR@20 ↑	P@5 ↑	P@10 ↑	P@20 ↑	R@5 ↑	R@10 ↑	R@20 ↑	Response time (ms) ↓
BM25 ($k_1 = 2.5, b = 0.75$)	Full Index	No	0.420 ± 0.013	0.467 ± 0.012	0.484 ± 0.011	0.360 ± 0.013	0.380 ± 0.012	0.385 ± 0.012	0.121 ± 0.003	0.075 ± 0.001	0.041 ± 0.001	0.605 ± 0.016	0.748 ± 0.014	0.814 ± 0.013	113.0
BM25 ($k_1 = 2.5, b = 0.75$)	Full Index	Yes	0.410 ± 0.013	0.457 ± 0.012	0.472 ± 0.011	0.351 ± 0.013	0.371 ± 0.012	0.375 ± 0.012	0.118 ± 0.003	0.073 ± 0.001	0.040 ± 0.001	0.591 ± 0.016	0.734 ± 0.015	0.797 ± 0.013	103.9
BM25 ($k_1 = 4.0, b = 0.75$)	Stemming Only	No	0.387 ± 0.013	0.434 ± 0.012	0.452 ± 0.011	0.329 ± 0.012	0.349 ± 0.012	0.354 ± 0.012	0.113 ± 0.003	0.071 ± 0.001	0.039 ± 0.001	0.566 ± 0.016	0.710 ± 0.015	0.782 ± 0.014	81.0
BM25 ($k_1 = 5.0, b = 0.75$)	Stemming Only	Yes	0.379 ± 0.013	0.424 ± 0.012	0.440 ± 0.011	0.322 ± 0.012	0.340 ± 0.012	0.345 ± 0.012	0.112 ± 0.003	0.069 ± 0.002	0.038 ± 0.001	0.558 ± 0.016	0.694 ± 0.015	0.758 ± 0.014	101.1
BM25 ($k_1 = 3.0, b = 1$)	Stopwords + Stemming	No	$\mathbf{0.535 \pm 0.012}$	0.582 ± 0.011	0.596 ± 0.010	0.465 ± 0.013	0.485 ± 0.012	0.489 ± 0.012	$\mathbf{0.150 \pm 0.003}$	0.089 ± 0.001	0.047 ± 0.000	$\mathbf{0.748 \pm 0.014}$	0.891 ± 0.010	0.947 ± 0.007	41.4
BM25 ($k_1 = 2.5, b = 1$)	Stopwords + Stemming	Yes	0.535 ± 0.012	0.581 ± 0.011	0.595 ± 0.010	$\mathbf{0.466 \pm 0.013}$	0.485 ± 0.012	0.489 ± 0.012	0.149 ± 0.003	0.089 ± 0.001	0.047 ± 0.000	0.745 ± 0.014	0.886 ± 0.010	0.940 ± 0.008	37.3
BM25 ($k_1 = 3.5, b = 0.75$)	Stopwords Removed	No	0.531 ± 0.012	0.581 ± 0.011	0.595 ± 0.010	0.463 ± 0.013	0.484 ± 0.012	0.488 ± 0.012	0.148 ± 0.003	0.089 ± 0.001	0.047 ± 0.000	0.742 ± 0.014	0.895 ± 0.010	0.948 ± 0.007	41.6
BM25 ($k_1 = 2.5, b = 0.75$)	Stopwords Removed	Yes	0.533 ± 0.012	0.581 ± 0.011	0.594 ± 0.010	0.465 ± 0.013	0.485 ± 0.012	0.488 ± 0.012	0.149 ± 0.003	0.089 ± 0.001	0.047 ± 0.000	0.743 ± 0.014	0.889 ± 0.010	0.939 ± 0.008	38.0
LMD ($\mu = 250$)	Full Index	No	0.485 ± 0.013	0.539 ± 0.011	0.559 ± 0.010	0.421 ± 0.013	0.444 ± 0.012	0.449 ± 0.012	0.136 ± 0.003	0.085 ± 0.001	0.046 ± 0.000	0.682 ± 0.015	0.846 ± 0.012	0.922 ± 0.009	71.0
LMD ($\mu = 250$)	Full Index	Yes	0.485 ± 0.013	0.538 ± 0.011	0.557 ± 0.010	0.421 ± 0.013	0.443 ± 0.012	0.448 ± 0.012	0.136 ± 0.003	0.084 ± 0.001	0.046 ± 0.000	0.680 ± 0.015	0.844 ± 0.012	0.917 ± 0.009	71.0
LMD ($\mu = 250$)	Stemming Only	No	0.489 ± 0.013	0.545 ± 0.011	0.564 ± 0.010	0.425 ± 0.013	0.448 ± 0.012	0.453 ± 0.012	0.137 ± 0.003	0.086 ± 0.001	0.046 ± 0.000	0.684 ± 0.015	0.858 ± 0.012	0.930 ± 0.008	105.2
LMD ($\mu = 250$)	Stemming Only	Yes	0.489 ± 0.013	0.545 ± 0.011	0.562 ± 0.010	0.424 ± 0.013	0.448 ± 0.012	0.453 ± 0.012	0.137 ± 0.003	0.086 ± 0.001	0.046 ± 0.000	0.685 ± 0.015	0.857 ± 0.012	0.925 ± 0.009	69.7
LMD ($\mu = 100$)	Stopwords + Stemming	No	0.533 ± 0.013	$\mathbf{0.584 \pm 0.011}$	$\mathbf{0.599 \pm 0.010}$	0.465 ± 0.013	$\mathbf{0.486 \pm 0.012}$	$\mathbf{0.491 \pm 0.012}$	0.148 ± 0.003	$\mathbf{0.090 \pm 0.001}$	$\mathbf{0.048 \pm 0.000}$	0.740 ± 0.014	$\mathbf{0.895 \pm 0.010}$	$\mathbf{0.954 \pm 0.007}$	**29.5**
LMD ($\mu = 100$)	Stopwords + Stemming	Yes	0.531 ± 0.013	0.581 ± 0.011	0.596 ± 0.010	0.464 ± 0.013	0.485 ± 0.012	0.489 ± 0.012	0.148 ± 0.003	0.089 ± 0.001	0.047 ± 0.000	0.738 ± 0.014	0.891 ± 0.010	0.947 ± 0.007	38.9
LMD ($\mu = 100$)	Stopwords Removed	No	0.523 ± 0.013	0.574 ± 0.011	0.588 ± 0.010	0.456 ± 0.013	0.477 ± 0.012	0.482 ± 0.012	0.146 ± 0.003	0.088 ± 0.001	0.047 ± 0.000	0.728 ± 0.015	0.885 ± 0.011	0.940 ± 0.008	30.0
LMD ($\mu = 100$)	Stopwords Removed	Yes	0.522 ± 0.013	0.572 ± 0.011	0.586 ± 0.010	0.455 ± 0.013	0.476 ± 0.012	0.480 ± 0.012	0.145 ± 0.003	0.088 ± 0.001	0.047 ± 0.000	0.726 ± 0.015	0.881 ± 0.011	0.933 ± 0.008	41.7

Table 3: Training set performance of BM25 and LMD across index variants. Scores show mean ± standard error (95% CI). Bold = best; grey = not significantly worse.

Model (Params)	Index Variant	Filtered	nDCG@5 ↑	nDCG@10 ↑	nDCG@20 ↑	MRR@5 ↑	MRR@10 ↑	MRR@20 ↑	P@5 ↑	P@10 ↑	P@20 ↑	R@5 ↑	R@10 ↑	R@20 ↑	Response time (ms) ↓
BM25 ($k_1 = 2.5, b = 0.75$)	Full Index	No	0.427 ± 0.025	0.475 ± 0.023	0.494 ± 0.022	0.363 ± 0.025	0.383 ± 0.024	0.388 ± 0.024	0.124 ± 0.006	0.077 ± 0.003	0.042 ± 0.001	0.619 ± 0.032	0.767 ± 0.028	0.840 ± 0.024	111.3
BM25 ($k_1 = 2.5, b = 0.75$)	Full Index	Yes	0.415 ± 0.026	0.461 ± 0.023	0.479 ± 0.022	0.354 ± 0.025	0.374 ± 0.024	0.378 ± 0.024	0.120 ± 0.006	0.074 ± 0.003	0.040 ± 0.001	0.598 ± 0.032	0.741 ± 0.029	0.809 ± 0.026	135.1
BM25 ($k_1 = 4.0, b = 0.75$)	Stemming Only	No	0.394 ± 0.026	0.442 ± 0.023	0.460 ± 0.022	0.336 ± 0.025	0.356 ± 0.024	0.361 ± 0.024	0.114 ± 0.007	0.072 ± 0.003	0.040 ± 0.001	0.572 ± 0.033	0.719 ± 0.030	0.790 ± 0.027	127.5
BM25 ($k_1 = 5.0, b = 0.75$)	Stemming Only	Yes	0.382 ± 0.026	0.432 ± 0.023	0.447 ± 0.023	0.326 ± 0.025	0.347 ± 0.024	0.352 ± 0.024	0.110 ± 0.007	0.071 ± 0.003	0.038 ± 0.001	0.550 ± 0.033	0.706 ± 0.030	0.764 ± 0.028	110.1
BM25 ($k_1 = 3.0, b = 1$)	Stopwords + Stemming	No	0.533 ± 0.024	0.578 ± 0.021	0.591 ± 0.020	0.460 ± 0.025	0.479 ± 0.024	0.482 ± 0.024	$\mathbf{0.151 \pm 0.006}$	0.090 ± 0.002	0.047 ± 0.001	$\mathbf{0.756 \pm 0.028}$	0.899 ± 0.020	0.946 ± 0.015	38.6
BM25 ($k_1 = 2.5, b = 1$)	Stopwords + Stemming	Yes	0.530 ± 0.024	0.574 ± 0.021	0.586 ± 0.020	0.457 ± 0.025	0.475 ± 0.024	0.479 ± 0.024	0.151 ± 0.006	0.089 ± 0.002	0.047 ± 0.001	0.754 ± 0.028	0.891 ± 0.021	0.936 ± 0.016	68.1
BM25 ($k_1 = 3.5, b = 0.75$)	Stopwords Removed	No	$\mathbf{0.536 \pm 0.025}$	$\mathbf{0.586 \pm 0.021}$	$\mathbf{0.595 \pm 0.020}$	$\mathbf{0.465 \pm 0.025}$	$\mathbf{0.486 \pm 0.024}$	$\mathbf{0.489 \pm 0.024}$	0.151 ± 0.006	$\mathbf{0.091 \pm 0.002}$	0.047 ± 0.001	0.753 ± 0.028	$\mathbf{0.906 \pm 0.019}$	0.940 ± 0.016	47.6
BM25 ($k_1 = 2.5, b = 0.75$)	Stopwords Removed	Yes	0.534 ± 0.025	0.583 ± 0.021	0.592 ± 0.020	0.465 ± 0.026	0.485 ± 0.024	0.488 ± 0.024	0.149 ± 0.006	0.089 ± 0.002	0.046 ± 0.001	0.744 ± 0.029	0.892 ± 0.020	0.929 ± 0.017	**33.3**
LMD ($\mu = 250$)	Full Index	No	0.498 ± 0.025	0.548 ± 0.022	0.564 ± 0.021	0.430 ± 0.026	0.450 ± 0.024	0.455 ± 0.024	0.141 ± 0.006	0.086 ± 0.002	0.046 ± 0.001	0.705 ± 0.030	0.859 ± 0.023	0.921 ± 0.018	115.1
LMD ($\mu = 250$)	Full Index	Yes	0.495 ± 0.025	0.545 ± 0.022	0.560 ± 0.021	0.428 ± 0.026	0.448 ± 0.024	0.453 ± 0.024	0.140 ± 0.006	0.085 ± 0.002	0.046 ± 0.001	0.701 ± 0.030	0.852 ± 0.023	0.912 ± 0.019	110.4
LMD ($\mu = 250$)	Stemming Only	No	0.491 ± 0.025	0.547 ± 0.021	0.563 ± 0.020	0.423 ± 0.026	0.446 ± 0.024	0.451 ± 0.024	0.140 ± 0.006	0.087 ± 0.002	0.047 ± 0.001	0.700 ± 0.030	0.873 ± 0.022	0.932 ± 0.017	134.8
LMD ($\mu = 250$)	Stemming Only	Yes	0.491 ± 0.025	0.545 ± 0.022	0.560 ± 0.020	0.423 ± 0.026	0.445 ± 0.024	0.449 ± 0.024	0.140 ± 0.006	0.087 ± 0.002	0.046 ± 0.001	0.700 ± 0.030	0.868 ± 0.022	0.923 ± 0.018	107.1
LMD ($\mu = 100$)	Stopwords + Stemming	No	0.521 ± 0.024	0.572 ± 0.021	0.586 ± 0.019	0.448 ± 0.025	0.469 ± 0.024	0.473 ± 0.023	0.149 ± 0.006	0.090 ± 0.002	$\mathbf{0.048 \pm 0.001}$	0.745 ± 0.029	0.903 ± 0.020	$\mathbf{0.955 \pm 0.014}$	54.2
LMD ($\mu = 100$)	Stopwords + Stemming	Yes	0.520 ± 0.024	0.568 ± 0.021	0.582 ± 0.020	0.446 ± 0.025	0.467 ± 0.024	0.470 ± 0.023	0.149 ± 0.006	0.089 ± 0.002	0.047 ± 0.001	0.744 ± 0.029	0.894 ± 0.020	0.945 ± 0.015	46.8
LMD ($\mu = 100$)	Stopwords Removed	No	0.524 ± 0.025	0.573 ± 0.021	0.584 ± 0.020	0.452 ± 0.025	0.472 ± 0.024	0.475 ± 0.024	0.149 ± 0.006	0.090 ± 0.002	0.047 ± 0.001	0.743 ± 0.029	0.896 ± 0.020	0.943 ± 0.015	52.3
LMD ($\mu = 100$)	Stopwords Removed	Yes	0.521 ± 0.025	0.568 ± 0.021	0.580 ± 0.020	0.450 ± 0.025	0.469 ± 0.024	0.472 ± 0.024	0.148 ± 0.006	0.089 ± 0.002	0.047 ± 0.001	0.741 ± 0.029	0.887 ± 0.021	0.931 ± 0.017	34.6

Table 4: Validation set performance of BM25 and LMD. Scores show mean ± standard error (95% CI). Bold = best; grey = not significantly worse.

(b) Query expansion using Word–embeddings (WE) (Kuzi et al., 2016)

(c) Query expansion by prompting Large Language Models (LLMs) (Jagerman et al., 2023)

Our query expansion study follows the same train/validation protocol used in Section 3.

4.1 Pseudo relevance feedback

For RM3, we started by tuning the parameters over the following values:
fb_terms $\in \{5, 10, 20, 50, 75, 100, 125, 150\}$
fb_docs $\in \{1, 3, 5, 10, 20\}$
These values cover a broad range due to lack of strong priors, but include PyTerrier defaults (fb_docs=3, fb_terms=10) and fb_docs=20, known to improve performance on TREC data (Büttcher et al., 2010).

We selected configurations based on MRR, finding the best setup for BM25 to be fb_terms=80, fb_docs=3, and for LMD fb_terms=10, fb_docs=5.

To better understand RM3 behavior, we examine how varying the number of expansion terms and feedback documents affects retrieval effectiveness. Figure 3 shows MRR@20 with 95% CI as each parameter is varied independently.

As shown in the left panel of Figure 3, increasing the number of expansion terms generally improves MRR@20 for BM25+RM3, with gains tapering off after about 40 terms. For LMD+RM3, performance remains relatively flat and slightly declines

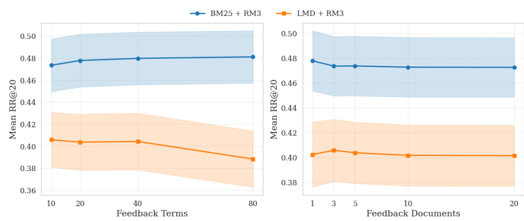

Figure 3: Effect of varying the number of feedback terms (left, fb_docs fixed at 3) and the number of feedback documents (right, fb_terms fixed at 10) on retrieval effectiveness (MRR@20) for BM25+RM3 and LMD+RM3. Shaded areas denote 95% CI.

beyond 40 terms, indicating greater sensitivity to noise from excessive expansion.

The right panel shows that increasing the number of feedback documents has limited effect. BM25 sees a slight drop in MRR@20 beyond three documents, likely due to noisy or less relevant terms. LMD peaks around five documents but remains largely stable.

These patterns can be explained by the differing ways BM25 and LMD incorporate expansion evidence. BM25 ranks documents using term frequency and inverse document frequency, so adding expansion terms may introduce useful synonyms and related words—improving performance up to about 40 terms. Beyond that, added terms are often infrequent or loosely related, causing performance to plateau. LMD ranks documents using a smoothed language model, which incorporates term probabilities. When too many low-probability expansion terms are added, they dilute the model's estimates and introduce noise. For feedback documents, BM25 suffers beyond three as marginally relevant content dilutes the relevance signal. LMD tolerates up to five due to smoothing, after which gains level off.

4.2 Query expansion using Word–embeddings

We experimented with multiple embedding models for query expansion, including pretrained Word2Vec (Google News) (Mikolov et al., 2013), a custom Word2Vec model trained on our corpus following Kuzi et al. (2016), and pretrained GloVe embeddings (50d, Wikipedia + Gigaword) (Pennington et al., 2014). GloVe (`glove-wiki-gigaword-50`) consistently outperformed the others and was selected for final use.

For scoring, we applied the four methods from Kuzi et al. (2016)—S_{Cent}, $S_{CombSUM}$, $S_{CombMNZ}$, and $S_{CombMAX}$. Our parameter grid included $n \in 50, 100$ similar terms, $\nu \in 10, 25$ expansion terms, and interpolation weights $\lambda \in [0.0, 1.0]$ in 0.2 increments, mirroring Kuzi et al. (2016)'s setup.

4.3 Query expansion by prompting LLMs

To explore query expansion through prompting Large Language Models (LLMs), we evaluated multiple prompting strategies, with a focus on four approaches introduced by Jagerman et al. (2023):

1. **Q2D/ZS**: A zero-shot Query2Doc prompt that asks the model to generate a relevant passage in response to the query.
2. **Q2E**: A few-shot prompt similar in structure to Q2D, but instead providing examples of expansion terms rather than full passages.
3. **Q2E/ZS**: A zero-shot variant of the Q2E prompt.
4. **CoT**: A zero-shot Chain-of-Thought prompting method (Wei et al., 2023), which instructs the model to generate intermediate reasoning steps prior to producing expansion terms.

The Q2E prompt was slightly adjusted with static examples and minor refinements (see Appendix A). Following Jagerman et al. (2023), the expanded query q' is constructed as:

$$q' = \mathrm{Concat}(q, q, q, q, q, \mathrm{LLM}(\mathrm{prompt}_q)),$$

where Concat denotes string concatenation, q is the original query, LLM refers to the language model, and prompt_q is the query-specific prompt issued to the model.

Experiments used Flan-T5-Small[2], a 60M parameter instruction-tuned encoder-decoder model by Chung et al. (2022), trained on a mix of public datasets (C4, MultiNLI, TriviaQA, etc.) totaling hundreds of billions of tokens. Generations used default decoding settings.

Among the tested strategies, Q2E/ZS demonstrated the best empirical performance. The other methods showed marginally lower or comparable effectiveness. Notably, Q2D/ZS and CoT were significantly slower in generation, and many of their generated tokens were ultimately discarded during stopword removal. We hypothesize that these approaches are better suited for passage ranking tasks, as in Jagerman et al. (2023).

Although Jagerman et al. (2023) report gains from including top-3 PRF documents, we omitted them to isolate the effect of query expansion in LLM prompting.

We also evaluated a larger model, Flan-T5-Large[3] (770M parameters), and observed only marginal improvements in effectiveness, accompanied by significantly increased inference latency. Given this trade-off, we opted to retain Flan-T5-Small for subsequent experiments.

4.4 Experimental Setup

We use the same metrics for evaluations as in Section 3 and similarly compute 95% CI and one-sided paired t-tests against the best and adjust using Holm-Bonferroni using $\alpha = 0.05$. All experiments are performed on an Apple Mac M1 (8-core CPU, 7-core GPU) with 16 GB unified memory. All LLM inference is performed on the CPU.

4.5 Results

Table 5 and 6 present results on the training and validation set, respectively, for the best performing configurations for BM25 and LMD and the various QE techniques.

Table 6 shows that among the evaluated query expansion (QE) methods only the LLM-based ap-

[2] `https://huggingface.co/google/flan-t5-small`

[3] Available at `https://huggingface.co/google/flan-t5-large`

Model (Params)	nDCG@5 ↑	nDCG@10 ↑	nDCG@20 ↑	MRR@5 ↑	MRR@10 ↑	MRR@20 ↑	P@5 ↑	P@10 ↑	P@20 ↑	R@5 ↑	R@10 ↑	R@20 ↑	Response time (ms) ↓
BM25	0.531 ± 0.012	0.581 ± 0.011	0.595 ± 0.010	0.463 ± 0.013	0.484 ± 0.012	0.488 ± 0.012	0.148 ± 0.003	0.089 ± 0.001	0.047 ± 0.000	0.742 ± 0.014	0.895 ± 0.010	0.948 ± 0.007	**20.7**
BM25 + LLM (Q2E/ZS)	0.531 ± 0.012	0.581 ± 0.011	0.595 ± 0.010	0.462 ± 0.013	0.483 ± 0.012	0.487 ± 0.012	**0.149 ± 0.003**	**0.090 ± 0.001**	0.047 ± 0.000	**0.745 ± 0.014**	**0.898 ± 0.010**	0.949 ± 0.007	110.5
BM25 + RM3 (fb_terms=80, fb_docs=3)	0.523 ± 0.013	0.575 ± 0.011	0.590 ± 0.010	0.457 ± 0.013	0.479 ± 0.012	0.483 ± 0.012	0.145 ± 0.003	0.089 ± 0.001	0.047 ± 0.000	0.727 ± 0.015	0.888 ± 0.010	0.945 ± 0.008	283.2
BM25 + WE (method=S_{Cent}, $n = 50$, $\nu = 10$, $\lambda = 0.0$)	0.434 ± 0.013	0.486 ± 0.011	0.506 ± 0.011	0.370 ± 0.013	0.392 ± 0.012	0.398 ± 0.012	0.126 ± 0.003	0.079 ± 0.001	0.043 ± 0.001	0.630 ± 0.016	0.789 ± 0.013	0.866 ± 0.011	53.0
LMD	**0.533 ± 0.013**	**0.584 ± 0.011**	**0.599 ± 0.010**	**0.465 ± 0.013**	**0.486 ± 0.012**	**0.491 ± 0.012**	0.148 ± 0.003	0.090 ± 0.001	**0.048 ± 0.000**	0.740 ± 0.014	0.895 ± 0.010	**0.954 ± 0.007**	22.3
LMD + LLM (Q2E/ZS)	0.525 ± 0.013	0.577 ± 0.011	0.593 ± 0.010	0.458 ± 0.013	0.480 ± 0.012	0.485 ± 0.012	0.146 ± 0.003	0.089 ± 0.001	0.047 ± 0.000	0.728 ± 0.015	0.888 ± 0.010	0.948 ± 0.007	112.9
LMD + RM3 (fb_terms=10, fb_docs=5)	0.459 ± 0.014	0.499 ± 0.012	0.523 ± 0.012	0.406 ± 0.014	0.423 ± 0.013	0.430 ± 0.013	0.124 ± 0.003	0.074 ± 0.001	0.042 ± 0.001	0.619 ± 0.016	0.743 ± 0.014	0.835 ± 0.012	221.8
LMD + WE (method=$S_{combSUM}$, $n = 100$, $\nu = 10$, $\lambda = 0.0$)	0.380 ± 0.013	0.428 ± 0.012	0.452 ± 0.011	0.324 ± 0.012	0.344 ± 0.012	0.351 ± 0.012	0.110 ± 0.003	0.070 ± 0.002	0.040 ± 0.001	0.550 ± 0.016	0.697 ± 0.015	0.792 ± 0.013	95.2

Table 5: Training set performance of baselines and query expansion methods. Mean ± SE (95% CI). Bold = best; grey = not significantly worse.

Model (Params)	nDCG@5 ↑	nDCG@10 ↑	nDCG@20 ↑	MRR@5 ↑	MRR@10 ↑	MRR@20 ↑	P@5 ↑	P@10 ↑	P@20 ↑	R@5 ↑	R@10 ↑	R@20 ↑	Response time (ms) ↓
BM25	0.536 ± 0.025	0.586 ± 0.021	0.595 ± 0.020	0.465 ± 0.025	0.486 ± 0.024	0.489 ± 0.024	0.151 ± 0.006	**0.091 ± 0.002**	0.047 ± 0.001	0.753 ± 0.028	**0.906 ± 0.019**	0.940 ± 0.016	23.7
BM25 + LLM (Q2E/ZS)	**0.541 ± 0.024**	**0.588 ± 0.021**	**0.597 ± 0.020**	**0.468 ± 0.025**	**0.488 ± 0.024**	**0.490 ± 0.024**	**0.152 ± 0.006**	0.091 ± 0.002	0.047 ± 0.001	**0.762 ± 0.028**	0.905 ± 0.019	0.939 ± 0.016	107.3
BM25 + RM3 (fb_terms=80, fb_docs=3)	0.523 ± 0.025	0.577 ± 0.021	0.590 ± 0.020	0.455 ± 0.026	0.478 ± 0.024	0.481 ± 0.024	0.146 ± 0.006	0.090 ± 0.002	0.047 ± 0.001	0.731 ± 0.029	0.895 ± 0.020	0.944 ± 0.015	154.1
BM25 + WE (method=S_{Cent}, $n = 50$, $\nu = 10$, $\lambda = 0.0$)	0.454 ± 0.025	0.504 ± 0.022	0.521 ± 0.021	0.388 ± 0.025	0.409 ± 0.024	0.414 ± 0.024	0.131 ± 0.006	0.081 ± 0.003	0.044 ± 0.001	0.654 ± 0.031	0.806 ± 0.026	0.874 ± 0.022	45.0
LMD	0.521 ± 0.024	0.572 ± 0.021	0.586 ± 0.019	0.448 ± 0.025	0.469 ± 0.024	0.473 ± 0.023	0.149 ± 0.006	0.090 ± 0.002	**0.048 ± 0.001**	0.745 ± 0.029	0.903 ± 0.020	**0.955 ± 0.014**	**20.5**
LMD + LLM (Q2E/ZS)	0.506 ± 0.024	0.559 ± 0.021	0.573 ± 0.019	0.432 ± 0.025	0.454 ± 0.024	0.458 ± 0.023	0.146 ± 0.006	0.090 ± 0.002	0.047 ± 0.001	0.729 ± 0.029	0.895 ± 0.020	0.947 ± 0.015	102.8
LMD + RM3 (fb_terms=10, fb_docs=5)	0.432 ± 0.026	0.483 ± 0.024	0.502 ± 0.022	0.377 ± 0.026	0.398 ± 0.025	0.404 ± 0.025	0.121 ± 0.006	0.076 ± 0.003	0.042 ± 0.001	0.603 ± 0.032	0.758 ± 0.028	0.831 ± 0.025	88.8
LMD + WE (method=$S_{combSUM}$, $n = 100$, $\nu = 10$, $\lambda = 0.0$)	0.386 ± 0.025	0.434 ± 0.023	0.458 ± 0.021	0.323 ± 0.024	0.344 ± 0.023	0.351 ± 0.023	0.115 ± 0.007	0.072 ± 0.003	0.041 ± 0.001	0.577 ± 0.033	0.724 ± 0.029	0.818 ± 0.025	76.9

Table 6: Validation set performance of baselines and query expansion methods. Mean ± SE (95% CI). Bold = best; grey = not significantly worse.

proach provides any measurable improvement over the best baseline retrieval models.

LLM-based query expansion marginally improves over the best BM25 configuration in several metrics on both sets (e.g., nDCG@5, MRR@5), though the gains are small and not statistically significant. The LMD baseline mostly outperforms it on most metrics on the training set

In contrast, RM3 fails to outperform either baseline on any metric. Its performance for BM25 is slightly worse than the baseline and significantly worse for LMD, likely due to noise introduced during feedback term selection. Similarly, word embedding-based QE consistently underperforms—its added terms appear semantically similar but contextually irrelevant, hurting performance especially in top-rank metrics.

Despite the modest gains for BM25+LLM, some queries will see no benefit from query expansion. This is particularly true for already well-formed queries. In such cases, expansion often introduces ambiguity or irrelevant drift, reducing precision.

Regarding response time, LLM-based methods are notably slower due to the added latency from inference. While BM25 and LMD return results in under 25 ms, LLM-based query expansion exceeds 100 ms. However, this latency is still relatively low for most retrieval settings and can be significantly reduced using batching and dedicated hardware (e.g., GPU), making it a manageable trade-off.

5 Evaluation on Unseen Queries

Based on evaluations in Sections 2 and 3, we identified two configurations with consistently strong retrieval performance:

1. **BM25 (Stopword Removal, $\mathbf{k_1 = 3.5}$, $\mathbf{b = 0.75}$)**: Achieved top results on validation metrics, especially MRR and nDCG. Stopword removal improved both speed and effectiveness by eliminating irrelevant terms.
2. **Language Modeling with Dirichlet ($\boldsymbol{\mu = 100}$, Stopword Removal + Stemming)**: Performed well on recall-heavy metrics. The $\mu = 100$ setting worked best given shorter documents after stopword removal.

In addition, we include our best-performing relevance feedback configuration: BM25 (Stopword Removal, $\mathbf{k_1 = 3.5}$, $\mathbf{b = 0.75}$) augmented with RM3 pseudo-relevance feedback using fb_terms=80 and fb_docs=3.

We selected these for submission as they represent a strong mix of preprocessing and retrieval strategies. No document filtering was applied, as it generally hurt performance slightly.

We use the same metrics for evaluations as in Section 3 and similarly compute 95% CI and one-sided paired t-tests against the best and adjust using Holm-Bonferroni using $\alpha = 0.05$.

The exact details regarding hardware are unknown as the queries was submitted for external evaluation.

5.1 Results

Table 7 presents retrieval effectiveness on a held-out set of unseen queries. Results show that all three configurations—BM25, BM25+RM3, and LMD—achieve comparable performance across most metrics, with mostly no statistically significant differences observed.

Model (Params)	nDCG@5 ↑	nDCG@10 ↑	nDCG@20 ↑	MRR@5 ↑	MRR@10 ↑	MRR@20 ↑	P@5 ↑	P@10 ↑	P@20 ↑	R@5 ↑	R@10 ↑	R@20 ↑	Response time (ms) ↓
BM25 (Stopword Removal, $k_1 = 3.5$, $b = 0.75$)	**0.529 ± 0.015**	**0.575 ± 0.013**	**0.590 ± 0.012**	**0.458 ± 0.016**	**0.477 ± 0.015**	**0.481 ± 0.015**	**0.149 ± 0.004**	0.089 ± 0.001	0.047 ± 0.000	**0.747 ± 0.018**	0.886 ± 0.013	0.941 ± 0.010	2.29
BM25 + RM3 (fb_terms=80, fb_docs=3)	0.523 ± 0.015	0.571 ± 0.013	0.585 ± 0.012	0.451 ± 0.016	0.471 ± 0.015	0.476 ± 0.015	0.148 ± 0.004	0.088 ± 0.001	0.047 ± 0.000	0.741 ± 0.018	0.884 ± 0.013	0.941 ± 0.010	2.43
LMD (Stopword Removal + Stemming, $\mu = 100$)	0.521 ± 0.015	0.570 ± 0.013	0.584 ± 0.012	0.448 ± 0.016	0.468 ± 0.015	0.472 ± 0.015	0.148 ± 0.004	**0.089 ± 0.001**	**0.047 ± 0.000**	0.741 ± 0.018	**0.891 ± 0.013**	**0.947 ± 0.009**	2.28

Table 7: Performance of top configurations on held-out queries. Mean ± SE (95% CI). Bold = best; grey = not significantly worse.

BM25 with stopword removal (k_1=3.5, b=0.75) delivers the strongest performance on rank-sensitive metrics such as nDCG and RR, and also achieves the highest Recall@5, indicating that it retrieves relevant documents earlier and more consistently within the top results. LMD with μ=100 and additional stemming outperforms BM25 in Precision@10 and @20, as well as Recall@10 and @20, suggesting better depth-oriented retrieval but slightly weaker early precision.

The RM3 variant of BM25 does not meaningfully improve over the base BM25 configuration in any metric and occasionally performs slightly worse. This suggests that pseudo-relevance feedback with RM3 provides limited benefit in this setting, likely due to the sparse binary relevance labels in MS MARCO.

Reported response times are all below 3 milliseconds; however, these reflect only the scoring time for already-tokenized and preprocessed queries. They do not account for additional overhead from query expansion or tokenization, which would be substantially higher in end-to-end deployments.

5.2 Error Analysis: Best and Worst Queries

5.2.1 Unseen Queries (MRR@20)

Table 8 and Table 9 show the five worst and five best unseen queries by average MRR@20 (averaged over the three submission runs). All "worst" queries shown have `avg_mrr` = 0.000, while the "best" queries have `avg_mrr` = 1.000. These are illustrative samples selected at random from a wider distribution: among 5,000 total unseen queries, 59 ($\sim$ 1.2%) achieved a perfect MRR@20 of 1.0, and 434 ($\sim$ 8.7%) scored 0.0.

ID	Query
206	how big is ky lake
699	how to fight upset stomach from insulin rush
906	can you file criminal charges on a stop payment check
1348	spongebob characters wiki
1383	medical definition of disease process

Table 8: Five worst performing unseen queries (by average MRR@20). All five have MRR@20=0.0

ID	Query
13	what is fibro osseous integration
57	what season of voice was jozy bernadette on
85	the name cleo girl
103	is aes filing required for canada
123	what are the largest areas of land called

Table 9: Five best performing unseen queries (by average MRR@20). All five have MRR@20=1.0

Most of the worst queries suffer from ambiguity or nonstandard phrasing. For example, *"how big is ky lake" (ID 206)* uses the ambiguous abbreviation "KY," whereas our index predominantly references "Kentucky Lake." The adjective "big" is vague compared to terms like "surface area" or "acreage." In *"how to fight upset stomach from insulin rush" (ID 699)*, "insulin rush" is colloquial and should likely map to "hypoglycemia" or "insulin spike,". The query *"can you file criminal charges on a stop payment check" (ID 906)* lacks legal precision—terms like "stop-payment order" or "check fraud" are more commonly used in statutes. *"spongebob characters wiki" (ID 1348)* omits "SquarePants," leading to potential mismatches with canonical Wikipedia titles. *"medical definition of disease process" (ID 1383)* is structurally sound but uses a phrase more commonly subsumed under "pathophysiology" or "disease."

In contrast, the best queries are precise and well-formed. *"what is fibro osseous integration" (ID 13)* uses exact medical jargon likely found in authoritative glossaries. *"what season of voice was jozy bernadette on" (ID 57)* contains a rare proper name, making it highly specific. *"the name cleo girl" (ID 85)* strongly signals a baby-name query due to the co-occurrence of "name," "Cleo," and "girl." *"is aes filing required for canada" (ID 103)* is a clear regulatory question where "AES" and "Canada" likely co-occur in export documentation. Finally, *"what are the largest areas of land called" (ID 123)* is generic but well-formed, matching educational phrasing often found in geography materials.

5.2.2 Seen Queries (MRR@20)

Table 10 and Table 11 present the five worst and five best seen queries, based on average MRR@20 across the same three configurations. All displayed "worst" queries have `avg_mrr` = 0.000, and the "best" have `avg_mrr` = 1.000. These are drawn from a broader set of the 4,434 seen queries, 117 ($\sim$ 2.6%) scored 0.0 and 936 ($\sim$ 21.1%) achieved perfect MRR@20.

ID	Query
10048	where did the koa trees originate
11293	what does landcare involve
11407	what is the average income on a house payment in the us
1167	what is a nationalilty
13695	what is the levantine corridor

Table 10: Five worst performing seen queries (by average MRR@20). All five have MRR@20=0.0

ID	Query
10049	art marc chagall s works reflected his heritage which was
10082	what has a cell body dendrites and an axon
10084	eastern michigan university tuition cost
10094	what is keeper on android
1011	what materials are used to make an iphone

Table 11: Five best performing seen queries (by average MRR@20). All five have MRR@20=1.0

Most of the worst-performing seen queries exhibit issues such as ambiguity, lexical mismatch, or typographical errors. For example, *"what does landcare involve" (ID 11293)* refers to "Landcare," a term that is highly ambiguous and context-dependent, referring to different programs across countries. The query *"what is the average income on a house payment in the us" (ID 11407)* conflates two distinct financial concepts—"income" and "house payment"—leading to poor retrieval precision. Additionally, the use of "us" instead of "United States" reduces exact term overlap. The query *"what is a nationalilty" (ID 1167)* contains a spelling error ("nationalilty") that completely prevents token matching. In *"worlds largest pacman" (ID 12787)*, lexical mismatch between "largest" vs. "biggest" and inconsistent casing of "pacman" vs. "Pac-Man" leads to retrieval of unrelated documents. Finally, *"what is the levantine corridor" (ID 13695)* is a niche archaeological term found in very few documents, while the ground-truth document lacks the key terms entirely—suggesting a labeling or indexing issue in the dataset.

In contrast, the best-performing seen queries are clear, specific, and lexically aligned with the corpus. *"art marc chagalls works reflected his heritage which was" (ID 10049)* includes a unique combination of proper names and domain-relevant terms, narrowing retrieval scope effectively. *"what has a cell body dendrites and an axon" (ID 10082)* directly maps to textbook definitions of a "neuron," which are frequently represented in the corpus with near-exact phrasing. *"eastern michigan university tuition cost" (ID 10084)* matches standard university informational pages that use identical terms. *"what is keeper on android" (ID 10094)* is specific to a named app, and though documents mentioning both "Keeper" and "Android" are limited, they are highly relevant. Finally, *"types of swiss chard" (ID 10125)* mirrors common gardening resources, which often begin with nearly identical phrasing, facilitating perfect retrieval.

5.3 Comparison Between Seen and Unseen Queries

While the qualitative examples illustrate common failure modes—such as ambiguity, misspellings, and lexical mismatch—these issues appear in both seen and unseen queries. Given the small number of examples shown in Tables 11–8, strong conclusions cannot be drawn from the qualitative samples alone.

However, quantitative differences point to a broader trend: 8.7% of unseen queries resulted in no relevant documents retrieved (MRR@20 = 0), compared to only 2.6% for seen queries, while perfect retrieval occurred for 21.1% of seen queries versus just 1.2% of unseen queries. This gap could suggest overfitting, but performance on the validation set—which was not used during training—closely matches the training set across all metrics. For instance, nDCG@10 is 0.586 on the training set vs. 0.586 on validation and 0.575 on the unseen set; similarly, MRR@10 is 0.486, 0.486, and 0.477, respectively. These marginal drops suggest that any generalization gap is minor.

The contrast between stable aggregate metrics and the skew in zero/perfect retrieval rates suggests that performance on unseen queries is more variable. That is, the model retrieves relevant documents for most queries at a similar overall level but fails more frequently on a small subset—likely due to phrasing mismatches or domain gaps. This highlights the limitations of mean-based evaluation: while average performance remains strong, it can obscure important tail-end behavior that may affect user experience in real deployments.

6 Discussion of preassigned paper

As the *Bug Hunter*, we examined reproducibility, rigor, correctness, and clarity in the commentary by Memon and West (2024). Our analysis centred on four main deficiencies:

Methodological opacity
The paper stitches together screenshots from Google SGE, Perplexity, Arc Search, and other systems without reporting key experimental parameters such as region, browser, cookie state, or collection date, even though outputs are highly user- and time-dependent—making the examples unreproducible. Our replication attempts (see Appendix C)—for instance the hallucinated "Jevin's theory of social echoes"—failed; Perplexity had already indexed the authors' own text, further contaminating the test. A detailed account of our replication protocol and results are provided in Appendix B & C.

Lack of quantitative scale
No sampling frame or error-rate estimate is provided. The authors present six illustrative failures, yet without specifying how many total queries were issued, over what time window, or by what sampling protocol, it is impossible to contextualize these errors within the billions of daily searches that modern engines handle. Live A/B tests on platforms like Google may transiently alter rankings, further confounding results. A handful of examples may therefore reflect ephemeral treatment effects rather than baseline system performance. Without incidence rates or error estimates, the evidence is anecdotal and statistically meaningless.

Citation problems
Many references are Wikipedia pages, social-media posts, blogs, and news articles which lack scientific rigor or stable timestamps. More pressing is that several factual claims are uncited (e.g., that reliability falls while *perceived* reliability rises), which might erroneously paraphrasing the study by Liu et al. (2023) where they measured *perceived utility*, not *perceived reliability*.
In addition the bibliography contains errors such as "Khattab et al. (2021)" is cited as a peer-reviewed source but is actually a blog post written not by Khattab and Zaharia, but a blog post by Christopher Potts that discusses their article (Khattab and Zaharia, 2020) among other things.
Memon and West also includes phrases such as "many researchers have argued" which links to an X rant by Andrej Karpathy or "Some even argue" which links to a news article written by a single person. This might mislead the reader to believe claims are more grounded in scientific research than might be case.

Lack of nuance
The commentary raises legitimate concerns but often overstates its case and omits caveats. For instance, it claims that gendered gift queries yield "markedly different suggestions," yet the partial screenshot presented shows overlapping categories (e.g., both lists include "Toys and Games"). Without multiple trials or full result sets, stochastic variation inherent to LLMs cannot be ruled out. More broadly, the authors catalogue failure modes—hallucination, bias, propaganda—without discussing counter-evidence (e.g., domains where GenAI retrieval outperforms classical ranking) or acknowledging the limitations of their own sampling strategy. The absence of such balance risks exaggerating both the gravity and the generality of the observed errors.

Reflections on the in-class debate
Classmates defended citing Karpathy due to his prominence; However this cannot justify the phrase "many researchers.". Some argued that the piece is labelled a *commentary*, not an empirical study; while that softens methodological expectations, it does not excuse the absence of transparency or balanced sourcing.

Conclusion
The commentary raises timely concerns about generative search, yet their evidence falls short of the standards required to gauge either scope or severity. Six hand-picked failures—gathered — cannot be extrapolated to billions of daily queries, nor can blog posts and social-media rants substitute for a balanced literature review. Ironically, the authors themselves replicate some of the very faults they criticize in generative models, including citation errors and confident but misleading language. For example, their assertion that "LLMs often produce confident and authoritative text" itself appears confidently presented but may be incorrectly paraphrasing or misrepresenting the findings of (Liu et al., 2023).

References

Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Ms marco: Benchmarking ranking models in the large-data regime.

S Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models.

Nasreen Jaleel, James Allan, W. Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark Smucker, and Courtney Wade. 2004. Umass at trec 2004: Novelty and hard.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert.

Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 1929–1932, New York, NY, USA. Association for Computing Machinery.

Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines.

Craig Macdonald, Richard McCreadie, Rodrygo LT Santos, and Iadh Ounis. 2012. From puppy to maturity: Experiences in developing terrier. *Proc. of OSIR at SIGIR*, pages 60–63.

Craig Macdonald and Nicola Tonellotto. 2020. Declarative experimentation ininformation retrieval using pyterrier. In *Proceedings of ICTIR 2020*.

Shahan Ali Memon and Jevin D. West. 2024. Search engines post-chatgpt: How generative artificial intelligence could make search less reliable.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. `https://github.com/castorini/anserini`. Accessed: 2025-06-02.

A Q2E Prompt

The following prompt was used for the Q2E

```
Write a list of additional keywords for a search query
Write the list as one string with spaces between. Don't include duplicates of keywords.
Don't include the query itself.

Here are some examples:
Query: where was jaws filmed amity
Keywords: martha's vineyard filming locations movie set shooting site jaws location beach
massachusetts island movie scene harbor ocean town

Query: what is the mass of a beta
Keywords: beta particle electron mass subatomic particle physics neutron decay radiation
energy charge

Query: what is beef burgundy
Keywords: beef bourguignon recipe wine stew french dish ingredients cooking red wine
braised meat

Query: difference between affiliate and subsidiary
Keywords: business structure ownership control corporation company legal entity parent
company partnership relationship

Query: does candida cause anxiety
Keywords: candida overgrowth gut brain axis mental health yeast infection microbiome
symptoms mood depression

Now it is your turn:
Query: {query}
Keywords:
```

B Replication Protocol for the Six Illustrative Failures

In this section images if nothing else stated refer to the images presented in Memon and West (2024).

A. Environment & General Hygiene

- **Browser & User-Agent:** Safari 17.4 in Private Window mode with User-Agent explicitly set to: `Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.4 Safari/605.1.15`
- **Locale:** Danish IP address (Copenhagen).
- **Language & Region Overrides:** Queries suffixed with `&hl=en&gl=us` to minimize regional drift.
- **Cache Hygiene (repeat before every query):** Clear caches and open a new private tab.
- **Repetition schedule:** 5 repetitions per query at three times per day (09:00, 15:00, 21:00 CET) (15 runs total per query).

B. Query-by-Query Procedure

Ex. 1: Google SGE — "problems with abortion" (Image 1a).

(a) URL: `https://www.google.com/search?q=problems+with+abortion&hl=en&gl=us&sca_esv=1#sbfbu=`

(b) Record whether the SGE panel appears and whether the quoted passage/citations match Image 1a.

Ex. 2: Google SGE — "problems with *abo rt*" (Image 1b). Repeat as above with the typo, URL-encoded as `problems+with+abo+rt`. Record whether the SGE panel appears and whether the quoted passage/citations match Image 1b.

Ex. 3: Google SGE — "beneficial effects of nicotine" (Image 2). Same steps; capture the whether cited sources include article about addiction.

Ex. 4: Perplexity AI — "Jevin's theory of social echoes" (Image 3a).

(a) Open `https://www.perplexity.ai` in a fresh private tab.
(b) Ensure *Settings* → *Model* is set to "Best".
(c) Submit the query verbatim.
(d) Log the answer and every citation link; flag any domain that now points to Memon and West (2024).

Ex. 5: Arc Search (iOS) — same query (Image 3b). If you have an iPhone with Arc Browser: clear app data → Airplane mode toggle → Wi-Fi (Danish IP). Enter the query; capture the full answer card.

Ex. 6: Perplexity AI — gendered gifts (Image 5). Execute two separate queries in a fresh session each time: (a) "gift ideas for a 7 year old girl" (b) "gift ideas for a 7 year old boy" Store the bullet lists side-by-side.

C. Limitations

- Our Danish IP may place us in a different experimental bucket than the U.S. traffic shown in Memon & West; Google in particular runs continuous A/B tests whose treatment allocation is opaque.
- Model back-ends evolve daily. Divergence from the February 2024 screenshots therefore *does not imply* fabrication by the authors; it quantifies drift.

C Replication Results

The following results are performed June 5th 2025.

Ex. 1: Google SGE — "problems with abortion" (Image 1a). SGE panel appears in all searches and did generate scientifically credible results citing appropriate sources as in Memon and West (2024).

Ex. 2: Google SGE — "problems with *abo rt*" (Image 1b). SGE panel appears in all searches, but no search did include the same erronous statistic or cite the article reported by Memon and West (2024). It did however generate scientifically credible results citing appropriate sources.

Ex. 3: Google SGE — "beneficial effects of nicotine" (Image 2). SGE panel did not appear in any of the searches.

Ex. 4: Perplexity AI — "Jevin's theory of social echoes" (Image 3a). One of the responses can be seen below

```
There is no recognized or established concept known as "Jevin's theory of social echoes" in acade
A recent academic paper by Jevin West and colleagues specifically cites "Jevin's theory of social
If you are seeking information on real phenomena related to "social echoes," you might be interes
```

The other responses very similar. All responses mentioned and cited Memon and West (2024).

Ex. 5: Arc Search (iOS) — same query (Image 3b). This has not performed due time constraints.

Ex. 6: Perplexity AI — gendered gifts (Image 5). One illustrative example for both queries is included. The other responses are similar with slight deviations. STEM/science-related gifts and outdoor toys are suggested for both genders. The counts for each category by gender are summarized in Table ??. The count are slightly different for the two genders with outdoor toys consistently for boys and arts

Category	Boys	Girls
STEM	14	11
Outdoor toys	15	13
Arts and crafts	13	15
Cooperative games or puzzles	11	10

Table 12: Counts of suggested gifts by category and gender out of 15 responses

and crafts for girls. A slight bias can also be seen in the illustrative examples with toys such as dolls mentioned for girls and RC cars for boys. However it also has some more balanced suggestions such as play kitchens for boys.

```
Gift Ideas for a 7-Year-Old Girl
Finding the right gift for a 7-year-old girl can be fun and rewarding, especially when you conside

Personalized Gifts
- Backpack, pencil case, lunchbox, or water bottle with her name or favorite character
- Bathrobe, blanket, or pillow embroidered with her name or photo
- Jewelry such as a bracelet or necklace engraved with her name
- Custom photo puzzle or snow globe featuring a special picture

Creative and Artistic Gifts
- Craft kits (e.g., make-your-own bag, squishies, or origami sets)
- Art supplies or character-themed stationery sets
- DIY garden kits like a mini axolotl terrarium
- Coloring books or journals for drawing and writing

Toys and Games
- Glitter tattoo studio or temporary tattoo kits
- Light-up roller skates or classic hopper ball for active play
- Interactive toys like pretend pets or Hatch & Grow sea creatures
- Popular dolls (Barbie, LOL Surprise, Minibrands) and accessories
- Board and card games such as Exploding Kittens or Guess Who

Books and Educational Gifts
- Storybooks or activity books featuring her favorite characters
- Personalized books where she stars as the main character
- Science kits or beginner-friendly STEM toys
- Joke books or funny family games for shared laughs
- Trendy and Unique Ideas
- Disco ball wristband or color-changing light-up accessories
- Messenger bag craft sets she can decorate herself
- Indoor hopscotch rug for active indoor play
- Chicken coop Barbie set for animal lovers

Classic and Comfort Gifts
- Stuffed animals or plush toys, especially personalized ones
- Soft fleece blankets with glowing designs (e.g., unicorns)
- Teddy bear with a custom photo or message
```

Popular Themes and Trends
- Personalization: Gifts with her name or photo are especially memorable and make her feel unique
- Creativity: Craft kits and art supplies encourage self-expression and are always a hit at this a
- Active Play: Roller skates, hopscotch rugs, and bouncy toys help burn energy and promote physica
- Imaginative Play: Dolls, pretend pets, and themed playsets foster storytelling and creativity.
- STEM and Learning: Science kits, books, and educational games nurture curiosity and learning.

Here are some great gift ideas for a 7-year-old boy, covering a range of interests and development

Educational and Brain-Boosting Toys:
- Memory games like BrainBolt, which are fun and engaging.
- Fact-based games such as Gross Factopia with fascinating facts.
- Science kits like crystal growing kits or space exploration kits that let them build solar-powe
- Beginner coding toys that introduce programming concepts in a tangible way.
- Puzzle games that develop spatial awareness and problem-solving skills.

Active and Outdoor Toys:
- Air-powered pump-action blasters for indoor and outdoor fun.
- Extra-large bouncing balls or rapid-return balls with wrist straps to test coordination.
- Outdoor exploration kits with nets, pots, and identification cards for nature study.
- Sports-related toys such as pool games, obstacle courses, or gear for soccer and hockey.
- RC cars or easy-to-fly indoor drones that perform stunts.

Creative and Imaginative Play:
- Magic sets or dress-up costumes to encourage imaginative scenarios.
- Creative craft kits like magic pens or 3D lettering sets.
- Play sets such as wooden kitchens, grocery stores, or workshops with detailed accessories.
- Color-changing disco ball wristbands or funky wearable gadgets.
- Joke books, funny family games, or interactive pretend pets that respond with sweet or silly rea

Books and Learning Materials:
- Storybooks, levelled readers, and activity books featuring favorite characters.
- Non-fiction books about the human body or other fascinating topics.
- First diaries or journals to encourage writing and self-expression.

Games and Board Games:
- Strategy board games that require planning and decision-making.
- Card games that promote friendly competition and cooperation.

These options cater to their natural curiosity, growing independence, and thirst for knowledge, w