

Sift: Local Semantic Search without Query-Time Neural Inference

nullreference

Abstract

Semantic retrieval often adds a neural model call to each query. Sift moves semantic term expansion to index construction and serves queries from a local sparse index. On Cranfield and the BEIR SciFact test set, Sift had the lowest measured median query latency among eight evaluated systems, at 0.75–0.80 ms. Its nDCG@10 exceeded Terrier BM25 by 0.006 and 0.012, but remained below the best measured system on both collections. A four-document demonstration also retrieved English and Spanish matches when built with a multilingual model. These results support a narrow claim: Sift offers fast local retrieval with lexical-semantic matching and competitive relevance on limited evidence. They do not establish broad multilingual quality or production throughput.

1 Motivation

Exact-term search can miss a relevant document when its wording differs from the query. Dense retrieval addresses this lexical gap, but usually encodes each query at search time. Sift instead precomputes related-token contributions while building the index, then uses sparse scoring locally. This shifts model work out of the query path, at the cost of index-build work and an expanded index. BEIR shows why both retrieval quality and speed need measurement across tasks [1].

The measured comparison places Sift’s main advantage in the query path: it keeps semantic term expansion while avoiding query-time neural inference. On two CPU collections, it returned results faster than the tested systems, with relevance close to BM25 and below the strongest hybrid system. This is a system-level tradeoff, not evidence that Sift improves retrieval quality in general.

2 System

Sift tokenizes documents and builds a sparse term-document index. At index time, static token vectors select related terms and add weighted contributions to their postings. Query-time retrieval uses BM25-style sparse scoring over the expanded postings and retains exact term matches. In a local eight-document example, a query for “cat” ranked both the cat document and a document that used “kitten.” No neural model runs for each query. The model is needed when the index is built.

The benchmark uses Potion-base-8M, a static model distilled from an English BGE model [2]. Cross-language matching therefore requires a multilingual term model. Term expansion is not full query understanding. It relates individual terms, but does not by itself infer how syntax, negation, or word order changes the query’s meaning.

This design moves work to index construction. It can reduce query-time work, but it does not eliminate model dependencies: index construction loads the term model, and corpus-specific indexes must be built before serving.

3 Evaluation

Collections. Cranfield contains 1,400 documents and 225 queries. SciFact uses the BEIR test split, with 5,183 documents and 300 queries. The evaluation uses all judged queries. These two collections provide a small lexical collection and a scientific fact-verification collection. They do not represent the full range of retrieval domains.

Mechanism checks. Separate from the benchmarks, we tested Sift on its eight-document animal example and four English-Spanish documents. The bilingual corpus contained two parallel pairs: “A kitten is a young domestic cat” / “Un gatito es un gato joven” and “A puppy is a young dog” / “Un cachorro es un perro joven.” We built indexes with Potion-base-8M and Potion-multilingual-128M, using `--threshold 0.5 --k-expand 20 --stop-df 1.0`. These checks illustrate retrieval behavior. They are not relevance benchmarks.

Table 1: Relevance, query latency, and ingestion from the recorded CPU comparison. Latency is milliseconds.

Collection	System	nDCG@10	R@100	p50	p95	Build (s)
Cranfield	Sift	0.349	0.739	0.80	1.00	0.75
Cranfield	BM25 (Terrier)	0.343	0.745	4.25	5.20	1.12
Cranfield	SPLADE	0.359	0.775	37.50	43.08	106.25
Cranfield	Weaviate hybrid	0.370	0.768	20.46	23.63	46.22
SciFact	Sift	0.696	0.926	0.75	1.01	2.72
SciFact	BM25 (Terrier)	0.684	0.926	4.20	4.90	1.49
SciFact	SPLADE	0.708	0.949	37.41	42.95	379.67
SciFact	Weaviate hybrid	0.723	0.955	20.82	23.58	176.18

Systems. The comparison includes Terrier BM25, E5-small-v2, BGE-small-en-v1.5, SPLADE [4], and Weaviate vector, BM25, and hybrid modes. Neural systems use the recorded model revisions. Weaviate runs in Docker Desktop. Sift uses its release binary. Comparisons use the recorded model revisions and benchmark settings.

Measures and timing. We report nDCG@10 and Recall@100 for relevance. Query latency is application-level latency, including each adapter call. One warmup pass precedes two measured passes. Queries run one at a time in shuffled order, and relevance uses the first measured pass. The report aggregates per-query latency as p50 and p95. Ingestion is one full index build with model and dataset files already local. Measurements ran on an Apple M1 Ultra with 20 CPU cores and 128 GiB of memory. No system used a GPU.

4 Results

Sift had the lowest measured p50 and p95 query latency on both collections. Its nDCG@10 exceeded Terrier BM25, while Weaviate hybrid had the highest nDCG@10 on both. The table shows the measured speed-relevance tradeoff.

The local examples confirm the mechanism and its limits. With Potion-base-8M, “cat” retrieved the English kitten document, but the bilingual queries did not retrieve across languages. With Potion-multilingual-128M [3], “cat” ranked the English kitten document first and the Spanish gatito document second. “Gato” ranked the Spanish gatito document first and the English kitten document second. This is a four-document demonstration, not evidence of multilingual retrieval quality at scale. On the animal example, “not cat” still returned the cat and kitten documents. Term expansion alone does not provide reliable natural-language negation.

On Cranfield and SciFact, Sift’s nDCG@10 trailed Weaviate hybrid by 0.021 and 0.027. Its SciFact index took 2.72 s to build, compared with 1.49 s for Terrier BM25. Its latency advantage therefore came with relevance and dataset-dependent ingestion tradeoffs.

5 Limits

These measurements support a system-level tradeoff, not a general ranking claim. Only two collections were measured. Queries ran serially on one machine, with no concurrent load test. The benchmark reports point estimates without confidence intervals. We therefore cannot determine whether small relevance differences are stable across query samples, machines, model revisions, or Sift configurations. The results also do not compare index size, memory use, or operational cost across every system.

6 Conclusion

Sift paired sub-millisecond median query latency with relevance near the tested BM25 baseline on two CPU collections. A separate toy example showed related English terms matching locally, and showed cross-language matches with a multilingual model. The benchmarks support the speed claim, not broad relevance or multilingual claims. Term expansion also leaves query composition as a separate problem.

References

- [1] N. Thakur, N. Reimers, A. Rüchlé, A. Srivastava, and I. Gurevych. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *NeurIPS Datasets and Benchmarks*, 2021. <https://arxiv.org/abs/2104.08663>.
- [2] Minish Lab. potion-base-8M model card. Model distilled from BAAI/bge-base-en-v1.5. <https://huggingface.co/minishlab/potion-base-8M>.
- [3] Minish Lab. potion-multilingual-128M model card. Static multilingual model. <https://huggingface.co/minishlab/potion-multilingual-128M>.
- [4] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. <https://arxiv.org/abs/2109.10086>.