Semantic Recall for Vector Search

Leonardo Kuffo
CWI
Amsterdam, The Netherlands
lxkr@cwi.nl

Ioanna Tsakalidou
EPFL
Lausanne, Switzerland
ioanna.tsakalidou@epfl.ch

Roberta De Viti
MPI-SWS
Saarbrücken, Germany
rdeviti@mpi-sws.org

Albert Angel
Google
Zurich, Switzerland
albertangel@google.com

Jiří Iša
Google
Zurich, Switzerland
jisa@google.com

Rastislav Lenhardt
Google
Zurich, Switzerland
lenhardt@google.com

Abstract

We introduce *Semantic Recall*, a novel metric to assess the quality of approximate nearest neighbor search algorithms by considering only semantically relevant objects that are theoretically retrievable via exact nearest neighbor search. Unlike traditional recall, semantic recall does not penalize algorithms for failing to retrieve objects that are semantically irrelevant to the query, even if those objects are among their nearest neighbors. We demonstrate that semantic recall is particularly useful for assessing retrieval quality on queries that have few relevant results among their nearest neighbors—a scenario we uncover to be common within embedding datasets. Additionally, we introduce *Tolerant Recall*, a proxy metric that approximates semantic recall when semantically relevant objects cannot be identified. We empirically show that our metrics are more effective indicators of retrieval quality, and that optimizing search algorithms for these metrics can lead to improved cost-quality tradeoffs.

CCS Concepts

• **Information systems** → **Relevance assessment**.

Keywords

information retrieval, dense embeddings, vector search, approximate nearest neighbor search

ACM Reference Format:
Leonardo Kuffo, Ioanna Tsakalidou, Roberta De Viti, Albert Angel, Jiří Iša, and Rastislav Lenhardt. 2026. Semantic Recall for Vector Search. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3805712.3809894

1 Introduction

In recent years, information retrieval (IR) using vector embeddings has received unprecedented attention from both academia and industry. This surge is fueled by AI embedding models that effectively capture concepts from objects (e.g., text, images, videos) into high-dimensional vectors [17,18,30]. In high-dimensional vector spaces, the proximity of two vectors in terms of their semantic concepts is captured by distance metrics like Euclidean distance or inner product [11,27]. Consequently, modern IR systems switched to *semantic search*: user queries are transformed into vector embeddings, which are used to identify their nearest objects in a database using nearest neighbor search (NNS) algorithms [9,13,21].

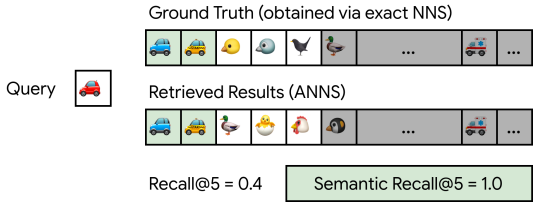

Figure 1: Semantic recall only considers the semantically relevant results in the ground truth that are theoretically retrievable via exact nearest neighbor search (in green).

Exact NNS is computationally expensive; to accelerate it, IR systems relax the constraints of search *exactness* and provide only *approximate* results. While these approximations are sufficient for most IR tasks, they raise a question: *How to measure search quality?* [31]. Ideally, developers would use end-to-end quality metrics during the development of approximate NNS (ANNS) algorithms (e.g., click-through rates, satisfaction surveys). However, this is generally impractical [3,8]; thus, academia and industry have standardized the use of *recall* for assessing retrieval quality of ANNS algorithms [4,13–15,20,22,24]. Given a dataset and a set of query embeddings, *recall* is defined as the proportion of retrieved nearest neighbors appearing in the queries' *ground truth*. A ground truth is a set of "true" nearest neighbors identified by ranking all dataset embeddings via brute-force (exact) NNS.

Despite its widespread use, recall fails to accurately capture the semantic quality of retrieved results [4,5,14,31,32]. This limitation stems from the reliance on ground truth defined via exact NNS, which reflects mathematical proximity rather than semantic relevance. As a result, low recall in ANNS can be misleading: failing to retrieve a mathematically close neighbor is penalized even when that neighbor is semantically irrelevant. Essentially, traditional recall penalizes the omission of "mathematical noise": irrelevant points that happen to lie closer to the query in the embedding space. This issue is particularly prevalent when queries have few relevant results, a scenario we find to be common in embedding datasets (§4.1). Consequently, traditional recall is misaligned with the goal of embedding-based retrieval: identifying semantically similar objects.

SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3809894

Table 1: Comparison of quality metrics for ANNS.

Criteria	Traditional Recall	Using curated neighbors	Semantic Recall	Tolerant Recall	Recall @k-ϵ	Relative Distance Error	R-Precision
Measures semantic relevance	✗	✓	✓	Partial	✗	✗	✓
Robust to model errors	✓	✗	✓	✓	✓	✓	✗
Hyperparameter-free	✓	✓	✓	✗	✗	✓	✓
No manual curation required	✓	✗	Partial	✓	✓	✓	Partial
Robust to dataset updates	✗	✗	✗	✓	✓	✓	✗
Interpretability of score	Easy	Easy	Easy	Fair	Fair	Hard	Fair

Occasionally, datasets include a curated ground truth where queries are manually matched to relevant objects. However, ANNS algorithms cannot retrieve *semantically* better neighbors than those identified by an exact NNS, as ANNS is constrained by the concepts captured in the embedding space [33]. For instance, in Figure 1, the ambulance emoji is an item relevant to the query. However, the embedding model failed to place this object close to the query in the vector space. Thus, an exact NNS misses it, and, by extension, any ANNS algorithm. Still, a curated ground truth may recognize the item as a relevant neighbor to the query. As a result, the use of curated ground truths may inadvertently penalize ANNS algorithms for limitations of the embedding model [5].

To address these issues, we introduce two novel metrics to evaluate retrieval quality of ANNS algorithms: semantic recall and tolerant recall. We define **semantic recall** as the fraction of *semantic neighbors* retrieved by an ANNS algorithm that are *theoretically retrievable* via exact NNS (Figure 1). Semantic neighbors are the subset of the ground truth that exhibit a semantic relationship with the query, as determined by an external *judge*. Furthermore, we introduce **tolerant recall**, a metric that approximates semantic recall by allowing the replacement of a true nearest neighbor with another retrieved result if their scores are *close* (i.e., within a given tolerance).

In the remainder of this work, we make the following contributions: (i) The definition of semantic recall, our novel proxy for search quality of ANNS algorithms (§ 2); (ii) A methodology for identifying the semantic neighbors of a set of queries (§ 2.1); (iii) The introduction of tolerant recall, a metric for scenarios where semantic neighbors cannot be obtained (§ 3); (iv) An empirical comparison of traditional recall, semantic recall, and tolerant recall (§ 4); (v) An empirical evaluation showing potential for improving the performance and reliability of ANNS systems by tuning search algorithms with our metrics (§ 4.3).

2 Semantic Recall

We define semantic neighbors as the subset of brute-force ground truth neighbors that are *semantically relevant* to the query. This semantic relevance is determined by an external *judge* (§ 2.1). Given a set of semantic neighbors (SN) taken from the top-k true nearest neighbors of a query, and a set of neighbors (R) retrieved by an ANNS algorithm, we define semantic recall (srecall) as the portion of R in SN: $srecall = \frac{|R \cap SN|}{|SN|}$. A high srecall indicates that the ANNS algorithm is effective at retrieving objects that are not just mathematically close, but also semantically meaningful.

Semantic recall addresses three critical limitations of traditional recall: (i) it does not penalize algorithms for missing mathematically closer but semantically irrelevant objects; (ii) it circumvents the limitations of the embedding model by focusing only on relevant objects that are present in the ground truth obtained via exact NNS, and (iii) it eliminates the need to configure threshold hyperparameters. This stands in contrast with metrics like *recall@k-ϵ* (used in ANN-benchmarks [2]), *robustness@k-γ* [32], and Relative Distance Error (RDE) [4,10], among others [31]. These metrics require thresholds that lack clear configuration heuristics and suffer from poor interpretability and portability across datasets. Moreover, RDE still penalizes ANNS algorithms for missing irrelevant neighbors. Finally, we believe that semantic recall is a useful metric for both end-user applications and developers. As a more accurate indicator of retrieval quality, it is a better proxy to measure whether an algorithmic improvement has any real impact on the end-user experience.

2.1 Finding Semantic Neighbors

To identify semantic neighbors, we employ a two-step process. First, we compute the ground truth of the query embeddings via exact NNS. Then, we classify each object in the ground truth as either semantically relevant or irrelevant to the query. This classification can be performed by human judges or by Large Language Models (LLMs), which excel at classification tasks [7,25]. The judging process with LLMs entails engineering a prompt tailored to the retrieval workload and the dataset (e.g., for Q&A corpora, a relevant document should answer the query text). Notably, this approach is significantly more efficient than dataset curation: instead of judging millions of (query, document) pairs to create a "gold standard" dataset, we only validate a reduced subset per query (i.e., the top-k results in the ground truth).

3 Tolerant Recall

In some scenarios (e.g., when one has only embeddings but not the underlying data), it may not be possible to identify semantic neighbors. To this end, we propose tolerant recall, a metric that allows a retrieved result to replace a true nearest neighbor when measuring search quality, if the distance of the query from this result is *close* to the distance of the query from the true neighbor.

Let us assume that we search for 20 ground-truth documents (G_{20}) and we retrieve 20 documents (R_{20}). Tolerant recall (trecall) identifies the largest subset $T \subset R_{20}$ s.t. every object $t_i \in T$ corresponds to a unique object $g_i \in G_{20}$, where either $g_i = t_i$, i.e., the document is in the ground truth, or, assuming inner product similarity, $t_i^{score} \geq g_i^{score} \cdot (1 - x\%)$, i.e., the scores of these documents are within a $x\%$ tolerance threshold. Then: $trecall@20 = \frac{|T|}{|G_{20}|}$.

As shown in §4.2, tolerant recall closely tracks semantic recall, as it is robust to small perturbations in the distance metric that alter the ranking of irrelevant neighbors, such as those introduced by

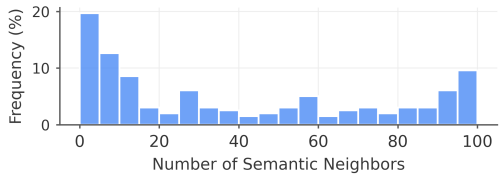

Figure 2: Distribution of semantically relevant neighbors per query in the MSMARCO dataset (250 queries).

Table 2: Inner product scores for semantic neighbors (SN) and non-semantic neighbors (non-SN) across queries.

	All Queries	SN < 20	20 ≤ SN < 80	SN ≥ 80
N. Queries	247	102	91	54
Avg. score (SN)	0.77 (± 0.02)	**0.77 (±0.03)**	0.76 (±0.02)	0.77 (±0.01)
Avg. score (non-SN)	0.73 (±0.03)	**0.71 (±0.02)**	0.74 (±0.02)	0.76 (±0.02)
Avg. score deltas (SN)	0.006 (±0.006)	**0.014 (±0.012)**	0.002 (±0.003)	0.001 (±0.001)
Avg. score deltas (non-SN)	0.002 (±0.003)	**0.001 (±0.002)**	0.001 (±0.002)	0.005 (±0.006)

quantization [10]. Furthermore, tolerant recall is more robust for monitoring frequently-updated datasets, whereas traditional recall can rapidly degrade if the ground truth is not updated alongside each database update. Additionally, for situations like a query having two high-scoring semantic results and the rest of the results being irrelevant (low-scoring results), tolerant recall captures the case of losing one of the semantic results. In contrast, *recall@k-ϵ* [2] may not register this as a loss, since it counts a result as correct if its score falls within a $(1+\epsilon)$ threshold of the k-th neighbor in the ground truth.

Table 1 compares various recall metrics, highlighting the benefits of semantic and tolerant recall for evaluating ANNS algorithms.

4 Evaluation

We evaluate our metrics on the MSMARCO corpus, which contains 8.83M text documents. We encode all documents into 3072-dimensional embeddings using the Gemini Embedding model [17]. Our query set consists of 250 general-knowledge questions, encoded with the same model. For each query, we compute the top-100 ground-truth neighbors via brute-force search using inner product similarity. Since the embeddings are normalized, inner product scores lie in $[-1, 1]$, with higher values indicating greater similarity. To identify semantic neighbors, we use Gemini 2.5 [6] as a judge to label each ground-truth neighbor as either *Relevant* or *Not Relevant* with respect to the query.

4.1 Analysis of Semantic Neighbors

Figure 2 shows the distribution of semantic neighbors per query as identified by Gemini. The distribution is bimodal: 40% of the queries have few relevant documents (0-15), including 3 queries with no relevant results, while another cluster appears near the upper end of the range (90-100 relevant documents). Further inspection reveals that queries in this latter group have more than 100 relevant documents in the corpus; the apparent upper bound is due to only using the top-100 neighbors in our experiments.

Table 2 compares the inner product scores of semantic neighbors (SN) and non-semantic neighbors (non-SN). SNs achieve higher average similarity scores across all query groups, with the largest gap appearing for queries with fewer than 20 SNs. Notably, SNs exhibit considerably larger and more variable score differences (deltas) between consecutive neighbors. In contrast, non-SNs exhibit consistently small score deltas, indicating that most irrelevant documents lie at nearly the same distance from the query embedding. This concentration increases sensitivity to ranking fluctuations among irrelevant ground-truth neighbors, increasing the likelihood that ANNS algorithms may omit such neighbors, leading to penalties under traditional recall [14]. These findings underscore the necessity for semantic recall: strictly penalizing algorithms for retrieving one irrelevant document over another is counterproductive.

Table 3: Summary of traditional, semantic, and tolerant recall on MSMARCO.

	Traditional Recall		Semantic Recall		Tolerant Recall	
	Avg.	Std.	Avg.	Std.	Avg.	Std.
All queries	0.863	0.177	**0.932**	0.144	**0.920**	0.125
Queries with < 20 SN	0.762	0.204	**0.903**	0.183	**0.859**	0.152
Queries with 20 to 80 SN	0.903	0.133	**0.937**	0.124	**0.941**	0.096
Queries with ≥ 80 SN	0.976	0.061	**0.978**	0.062	**0.991**	0.027

4.1.1 Cross validation. We cross-validated a subset of relevance judgments using Claude Haiku 4.5 as an independent judge outside the Gemini family. Specifically, we sampled 100 queries and their top-100 ground truth documents. The agreement rate between judges was 91.1% (Cohen's $\kappa = 0.82$). Looking into each category, agreement was 89.5% for "Relevant" and 93.3% for "Not Relevant". Only 2 queries fell below 70% agreement rate.

When the judges disagreed, Gemini more frequently labeled documents as "Relevant", while Claude adhered more strictly to the prompt criteria. Subsequent human inspection suggested that Gemini was more lenient, reasoning about ambiguities in query phrasing and document intent. For example, for the query *"when is the warm weather in Thailand"* and the document *"Thailand can best be described as tropical and humid for the majority of the country during most of the year."*, Claude labeled the document as not relevant due to the lack of specific temporal information, whereas Gemini labeled it as relevant. These findings highlight the importance of designing prompts that fit the use case of the application. For reference, our prompt is available in our code notebook [16].

4.2 Traditional, Semantic, and Tolerant Recall

To evaluate traditional, semantic, and tolerant recall, we ran top-100 searches using ScANN [13,22] (a partition-based ANNS index), with 8-bit quantization. Table 3 summarizes the results, and Figure 3 (top) shows the per-query distribution of traditional and semantic recall over 250 queries. Queries with few SNs are the most penalized by traditional recall, as near-equidistant irrelevant neighbors lead to missed ground-truth matches. In contrast, semantic recall better reflects the search quality perceived by end users (i.e., queriers), yielding a score of 0.903 compared to 0.762 for queries with few SN. This pattern holds in top-20 search, where traditional recall scores 0.869 and semantic recall 0.901. Tolerant recall with a 1% tolerance threshold closely approximates semantic recall (Figure 3, bottom). Unlike semantic recall, tolerant recall can be directly integrated into existing ANNS pipelines and remains usable in dynamic datasets.

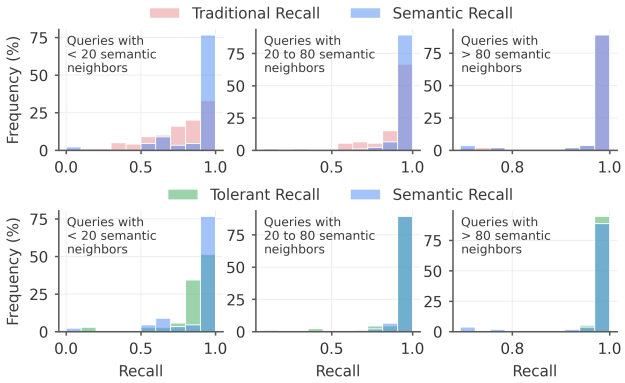

Figure 3: Per-query distributions of traditional (red), semantic (blue), and tolerant (green) recall on the MSMARCO dataset, grouped by semantic neighbor count (left to right).

Notably, the closeness of non-SNs also negatively affects traditional recall when evaluating quantization techniques. Figure 4 shows the relative errors between inner product scores computed with full precision and with 8-bit scalar quantization across MSMARCO, GloVe [23], and BigANN [28]. Even these small errors can reorder irrelevant neighbors, leading to penalties under traditional recall. Tolerant recall mitigates this effect by allowing replacements mostly within irrelevant results, and semantic recall avoids it by considering only relevant results, which are typically top-rankers in the ground truth.

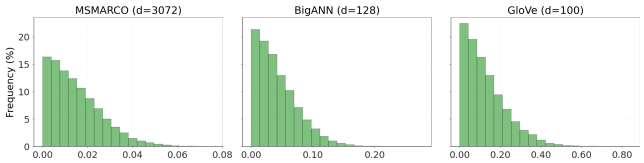

Figure 4: Distribution of the error % between scores computed with vectors at full precision and 8-bit quantization.

4.2.1 Comparison to R-Precision. Given the identified SNs, retrieval quality could be evaluated using R-precision. R-precision assumes that for a query with R relevant objects, an ideal retrieval system retrieves all R within the top-R results [35]. However, R-precision inadvertently penalizes the retrieval performance of ANNS algorithms for shortcomings in the embedding model. In fact, it assumes that the ground-truth ranking is aligned with semantic relevance, ignoring that irrelevant items may appear among top-ranked neighbors [1]. Although ground-truth rank and semantic relevance are negatively correlated for queries with fewer than 20 SNs (rank-biserial correlation of -0.72, p-value≈ 0), this correlation is not perfect. For instance, when retrieving $k = 5$ neighbors with $R = 3$ SNs, if the third SN is ranked fourth in the ground truth, R-precision counts this as an error. However, in this case, even an exact NNS would not have retrieved that document within the top-R. In contrast, semantic recall overcomes this limitation by allowing relevant documents to appear anywhere within the top-k results.

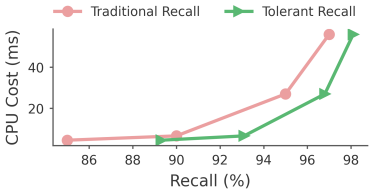

Figure 5: Recall vs cost: Cost rises sharply as recall increases.

4.3 Tuning for Semantic Recall

ANNS systems are commonly tuned to balance recall and search cost by selecting Pareto-optimal hyperparameters [2,9,14]. We show that using semantic and tolerant recall as tuning objectives yields a strictly better cost–quality tradeoff than tuning for traditional recall. We use Google Vizier [12] to tune the hyperparameters of ScaNN, minimizing search cost at a fixed target recall. We measure search cost as "bytes read", which closely aligns with the number of inner product computations [29]. Finally, we assess *performance* by measuring the CPU cost of query processing.

First, we tune ScaNN for a target traditional recall of 98% on MSMARCO, obtaining a configuration of Pareto-optimal hyperparameters that achieves 98% traditional recall, 98.69% tolerant recall, and 98.93% semantic recall. Then, we re-tune with tolerant recall as the objective, targeting 98.69%: this yields a configuration with the same tolerant recall at 5% lower cost. Similarly, tuning for semantic recall and targeting 98.93% reduces cost by 14% while preserving semantic recall. These gains arise because non-SNs are harder to retrieve: in partition-based indexes such as ScaNN, they require exploring additional partitions, increasing search effort.

Finally, we observe a significant opportunity to reduce costs by changing the target metric, e.g., from 95% traditional recall to 95% tolerant recall. The latter metric being a better representative of the real losses and always higher, enables this. In particular, tuning for 95% tolerant recall instead of 95% traditional recall reduces cost by ~25% on BigANN (1B vectors, 128 dimensions), ~5% on GloVe (1M vectors, 100 dimensions), and ~35% on MSMARCO. Notably, as shown in Figure 5, the cost grows sharply at higher recalls. As a result, whenever we manage to move slightly left on the recall-cost curve, we substantially reduce compute costs.

5 Generalizability

To demonstrate the generalizability of our metrics, we replicate our evaluation on a subset of the MIRACL dataset [34] consisting of 542,166 Thai-language documents. Documents and queries are embedded into 1024-dimensional vectors using Cohere Embed V3 [26]. The query set comprises 100 questions in Thai. For each query, we compute the top-20 ground truth neighbors via brute-force inner product search. We identify semantic neighbors (SNs) using Gemini 2.5 [6] as a judge. Figure 6 shows the distribution of SNs per query, which exhibits a power-law behavior similar to MSMARCO: 60% of the queries have at most 4 SNs. Similar to the MSMARCO dataset, SNs exhibit larger score deltas than non-SNs (0.05±0.03 in SN vs. 0.005±0.008 in non-SN). We then evaluate traditional, semantic, and tolerant recall with a 2% tolerance threshold using a FAISS IVF index [9] (3000 partitions, 8-bit scalar quantization), probing 0.5% of clusters. The average traditional, semantic, and tolerant recall

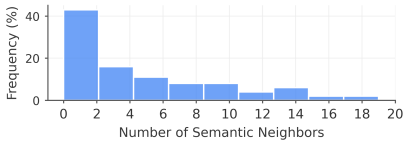

Figure 6: Distribution of the number of semantic neighbors per query in the MIRACL dataset.

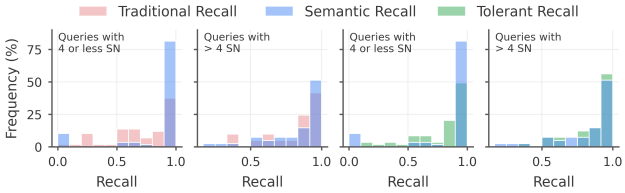

Figure 7: Per-query distribution of traditional, semantic, and tolerant recall on the MIRACL dataset.

are 0.75, 0.85, and 0.83, respectively. Figure 7 shows per-query recall distributions. As in MSMARCO, queries with few SNs exhibit low traditional recall, while semantic recall confirms that relevant results are retrieved. Tolerant recall serves as a middle ground, providing a better approximation of quality without requiring semantic judgments.

5.1 Choosing a tolerance threshold

Tolerant recall allows replacing irrelevant documents with retrieved documents with *similar* scores. Accordingly, the tolerance threshold should be sufficiently broad to enable such substitutions. We approximate this threshold using score differences among lower-ranked neighbors. In particular, we compute the relative difference between the $\frac{2}{3}k$-th and k-th neighbors from a top-k retrieval, normalized by the maximum ground-truth score. The tolerance threshold proxy is thus defined as $\frac{score_{2/3k} - score_k}{max\_score} * 100$ averaged across queries. Figure 8 shows that varying the threshold leads to different tolerant recall values. In both datasets, our proxy for the tolerance threshold closely matches the semantic recall metric. While this provides a useful starting point, there is potential for improvement, e.g., by estimating thresholds per query rather than per dataset. A conservative lower bound for the tolerance threshold can be determined from the relative error of scores resulting from the 8-bit compression of vectors, as 8 bits are sufficient to achieve almost perfect traditional recall in ANNS [10]. Finally, when SNs are available, the tolerance threshold can alternatively be obtained from the intersection of the tolerance threshold curve with semantic recall.

6 Discussion

Our results suggest that ANNS evaluation can relax strict mathematical precision in favor of improved search efficiency without compromising retrieval quality more than previously thought. In fact, semantic recall enables developers to distinguish between performance losses that affect relevant results and those that reshuffle irrelevant ones, supporting more meaningful cost—quality tradeoffs during algorithm design (e.g., when tuning quantization schemes or search parameters [4]). We further hypothesize that the long-tailed distributions of traditional recall reported in prior work on ANNS [4,14] stem from queries with few or no relevant neighbors.

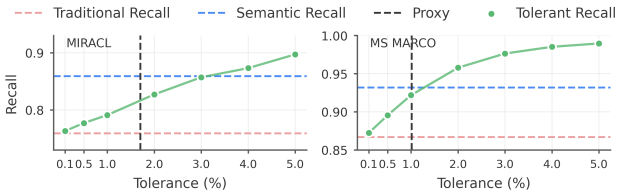

Figure 8: Effect of different thresholds on tolerant recall. Our proxy for setting a threshold closely follows semantic recall.

In such cases, hyperparameter tuning selects "costly" configurations that incur into additional effort to recover near-equidistant *irrelevant* neighbors, effectively optimizing for "noise". Thus, optimizing for traditional recall means optimizing for mathematical exactness at unnecessary extra retrieval cost.

Limitations. Semantic recall has practical limitations. Unlike traditional recall, it requires access to the underlying raw objects (text, images, ...) for judging. Also, the judging relies on the LLM's "knowledge": domain-specific datasets may still require human validation. However, judging only the top-k ground truth is more feasible than annotating entire datasets and more useful than pooled annotations. Furthermore, our approach adopts binary relevance judgments, which collapse uncertainty into a hard decision and delegate the confidence threshold to the judge. While richer graded judgments could capture uncertainty, we favor a binary definition for simplicity and reproducibility. Errors in judgment are unavoidable, and correlations between the judge and the evaluated system may occur in some scenarios. Finally, semantic recall is undefined when no relevant results exist (e.g., highly restrictive queries). Identifying these queries through the judging process is useful and worth inspection, as it may indicate regression in the embedding model. However, we argue that such queries should be excluded from retrieval-quality evaluation, as no meaningful answers are present. Conversely, when all results are relevant, semantic recall generalizes to recall, whereas tolerant recall enables replacements with objects that are likely also relevant due to their score proximity.

7 Conclusions

We introduced *semantic* and *tolerant* recall as alternatives to traditional recall that better capture retrieval quality in semantic search. Our results show that traditional recall penalizes ANNS algorithms for failing to retrieve near-equidistant but semantically irrelevant results, which are difficult to distinguish and costly to recover. By accounting for this structure, semantic and tolerant recall provide more faithful evaluation metrics and enable more meaningful cost–quality tradeoffs for algorithm design and tuning. Lastly, we validated the effectiveness of our metrics across different embedding models, ANNS algorithms, and data scales, demonstrating their applicability to vector search systems. We hope this work encourages the IR and Vector Search communities to incorporate evaluation metrics that better align with the objective of semantic search: retrieving relevant answers rather than the mathematically closest noise. As future work, we propose incorporating our metrics into early-termination mechanisms [4,20] and comparing our metrics across different ANNS algorithms (e.g., HNSW [19]). We provide a code notebook to reproduce our experiments [16].

References

[1] Mihir Agarwal, Ankit Garg, Neeraj Kayal, Kirankumar Shiragur, et al. 2026. On Strengths and Limitations of Single-Vector Embeddings. *arXiv preprint arXiv:2603.29519* (2026).

[2] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2020. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems* 87 (2020), 101374.

[3] Federico Cabitza, Andrea Campagner, and Valerio Basile. 2023. Toward a perspectivist turn in ground truthing for predictive computing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 6860–6868.

[4] Manos Chatzakis, Yannis Papakonstantinou, and Themis Palpanas. 2025. DARTH: Declarative Recall Through Early Termination for Approximate Nearest Neighbor Search. *Proceedings of the ACM on Management of Data* 3, 4 (2025), 1–26.

[5] Tingyang Chen, Cong Fu, Jiahua Wu, Haotian Wu, Hua Fan, Xiangyu Ke, Yunjun Gao, Yabo Ni, and Anxiang Zeng. 2025. Reveal Hidden Pitfalls and Navigate Next Generation of Vector Similarity Search from Task-Centric Views. *arXiv preprint arXiv:2512.12980* (2025).

[6] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261* (2025).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[8] Yifan Ding, Nicholas Botzer, and Tim Weninger. 2022. Posthoc verification and the fallibility of the ground truth. In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*. 23–29.

[9] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library. *IEEE Transactions on Big Data* (2025).

[10] Jianyang Gao, Yutong Gou, Yuexuan Xu, Yongyi Yang, Cheng Long, and Raymond Chi-Wing Wong. 2025. Practical and asymptotically optimal quantization of high-dimensional vectors in euclidean space for approximate nearest neighbor search. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–26.

[11] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[12] Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D. Sculley. 2017. Google Vizier: A Service for Black-Box Optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 1487–1495. doi:10.1145/3097983.3098043

[13] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*. PMLR, 3887–3896.

[14] Elias Jääsaari, Ville Hyvönen, Matteo Ceccarello, Teemu Roos, and Martin Aumüller. 2025. VIBE: Vector Index Benchmark for Embeddings. *arXiv preprint arXiv:2505.17810* (2025).

[15] Leonardo Kuffo, Elena Krippner, and Peter Boncz. 2025. PDX: A Data Layout for Vector Similarity Search. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–26.

[16] Leonardo Kuffo, Ioanna Tsakalidou, Roberta De Viti, Albert Angel, Jiří Iša, and Rastislav Lenhardt. 2026. Reproducibility: Semantic Recall for Vector Search. https://colab.research.google.com/drive/1cUnvdRP7CjeJvx5eaAzjA-J5d_d3oUk7. Google Colaboratory notebook.

[17] Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, et al. 2025. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891* (2025).

[18] Xianming Li, Aamir Shakir, Rui Huang, Julius Lipp, and Jing Li. 2025. ProRank: Prompt Warmup via Reinforcement Learning for Small Language Models Reranking. *arXiv preprint arXiv:2506.03487* (2025).

[19] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[20] Jason Mohoney, Devesh Sarda, Mengze Tang, Shihabur Rahman Chowdhury, Anil Pacaci, Ihab F Ilyas, Theodoros Rekatsinas, and Shivaram Venkataraman. 2025. Quake: Adaptive Indexing for Vector Search. *arXiv preprint arXiv:2506.03437* (2025).

[21] James Jie Pan, Jianguo Wang, and Guoliang Li. 2024. Survey of vector database management systems. *The VLDB Journal* 33, 5 (2024), 1591–1615.

[22] Yannis Papakonstantinou, Alan Li, Ruiqi Guo, Sanjiv Kumar, and Phil Sun. 2024. ScaNN for AlloyDB. (2024). https://services.google.com/fh/files/misc/scann_for_alloydb_whitepaper.pdf

[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[24] Jeffrey Pound, Floris Chabert, Arjun Bhushan, Ankur Goswami, Anil Pacaci, and Shihabur Rahman Chowdhury. 2025. MicroNN: An On-device Disk-resident Updatable Vector Database. In *Companion of the 2025 International Conference on Management of Data*. 608–621.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[26] Nils Reimers, Elliott Choi, Amr Kayid, Alekhya Nandula, Manoj Govindassamy, and Abdullah Elkady. 2023. *Introducing Embed v3.* Cohere. https://cohere.com/blog/introducing-embed-v3 Cohere Blog; published November 2, 2023.

[27] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[28] Harsha Vardhan Simhadri, George Williams, Martin Aumüller, Matthijs Douze, Artem Babenko, Dmitry Baranchuk, Qi Chen, Lucas Hosseini, Ravishankar Krishnaswamny, Gopal Srinivasa, et al. 2022. Results of the NeurIPS'21 challenge on billion-scale approximate nearest neighbor search. In *NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 177–189.

[29] Philip Sun, Ruiqi Guo, and Sanjiv Kumar. 2023. Automating Nearest Neighbor Search Configuration with Constrained Optimization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=KfptQCEKVW4

[30] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[31] Wenping Wang, Yunxi Guo, Chiyao Shen, Shuai Ding, Guangdeng Liao, Hao Fu, and Pramodh Karanth Prabhakar. 2023. Integrity and junkiness failure handling for embedding-based retrieval: A case study in social network search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3250–3254.

[32] Zikai Wang, Qianxi Zhang, Baotong Lu, Qi Chen, and Cheng Tan. 2025. Towards Robustness: A Critique of Current Vector Database Assessments. *arXiv preprint arXiv:2507.00379* (2025).

[33] Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. 2025. On the theoretical limitations of embedding-based retrieval. *arXiv preprint arXiv:2508.21038* (2025).

[34] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics* 11 (2023), 1114–1131.

[35] Keneilwe Zuva and Tranos Zuva. 2012. Evaluation of information retrieval systems. *International journal of computer science & information technology* 4, 3 (2012), 35.