Succeeding at Scale: Enterprise Retrieval Benchmark Construction and Index-Preserving Query Adaptation for Multi-Tenant Search

Prateek Jain [*1] **Shabari S Nair** [*2] **Ritesh Goru** [1] **Prakhar Agarwal** [3] **Ajay Yadav** [2]
Yoga Sri Varshan Varadharajan [2] **Constantine Caramanis** [2]

Abstract

Large-scale multi-tenant retrieval systems generate extensive query logs but lack curated relevance labels for effective domain adaptation, resulting in substantial underutilized "dark data." This challenge is compounded by the high cost of model updates, as jointly fine-tuning query and document encoders requires full corpus re-indexing, which is impractical in multi-tenant settings with thousands of isolated indices. We introduce DevRev-Search, a passage retrieval benchmark for technical customer support built via a fully automated pipeline. Candidate generation uses fusion across diverse sparse and dense retrievers, followed by an LLM-as-a-Judge for consistency filtering and relevance labeling. We further study and systematically evaluate index-preserving query-only adaptation strategies that fine-tune only the query-encoder while keeping the document indices fixed. Experiments on DevRev-Search, SciFact, and FiQA-2018 show that parameter-efficient fine-tuning of the query encoder delivers a remarkable quality–efficiency trade-off, enabling scalable and practical enterprise multi-tenant retrieval.

1. Introduction

The transition from lexical matching methods such as BM25 (Robertson & Zaragoza, 2009) to dense neural retrieval has revolutionized information discovery (Karpukhin et al., 2020). However, deploying bi-encoder architectures in multi-tenant enterprise environments presents two major challenges. First, the **Data Scarcity Bottleneck**: enterprise tenants possess proprietary "dark data" where relevance labels are unavailable, and standard benchmarks like BEIR (Thakur et al., 2021) fail to capture the noisy, heterogeneous nature of these domains. Second, the **Adaptation Latency Bottleneck**: symmetric fine-tuning of both encoders incurs a massive "Re-indexing Tax," as any document encoder update requires regenerating embeddings for the entire corpus, which is computationally prohibitive for platforms hosting thousands of tenants.

To address these challenges, we present a unified methodology for scalable dataset construction and efficient model adaptation. Our main contributions are as follows:

1. **DevRev-Search Benchmark** [1]: We construct a scalable pipeline that synthesizes training data without human annotation by pooling candidates from diverse retrievers and applying LLM-as-a-Judge filtering (Dai et al., 2023; Rahmani et al., 2024).

2. **Zero-Reindexing Adaptation**: We propose a new approach to fine-tuning dense retrieval models by only adapting the query encoder while keeping the document encoder (and indices) fixed.

3. **Parameter-Efficient Query Adaptation**: We comprehensively evaluate techniques such as LoRA (Hu et al., 2022), embedding transformations via linear and feed-forward projections, and selective fine-tuning of transformer layers, as efficient query-only fine-tuning strategies. We also conduct ablations over LoRA ranks, modules, and the number of unfrozen layers to find the best possible configurations.

Through extensive evaluation on DevRev-Search, SciFact (Wadden et al., 2022), and FiQA-2018 (Maia et al., 2018), we demonstrate that Parameter-Efficient query-only fine-tuning achieves robust domain adaptation, thus enabling tenant-specific models for multiple tenants at a fraction of the cost of full bi-encoder fine-tuning. In summary, we develop a unified framework for enterprise multi-tenant retrieval that combines a new scalable benchmark with a

[*]Equal contribution [1]DevRev, Austin, USA [2]The University of Texas at Austin, Austin, USA [3]DevRev, Bengaluru, India. Correspondence to: Prateek Jain <prateek.jain@devrev.ai>, Shabari S Nair <shabarisnair@utexas.edu>.

2nd Workshop on Connecting Low-rank Representations in AI at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

[1]The DevRev-Search dataset is made available at https://huggingface.co/datasets/devrev/search.

new, cost-effective domain adaptation approach based on query-only adaptation for dense retrieval. To the best of our knowledge, such an approach has not been explored in prior work, particularly in multi-tenant settings.

2. Related Work

Synthetic Data for Retrieval: Recent work leverages LLMs for synthetic data generation (Bonifacio et al., 2022; Wang et al., 2022; Dai et al., 2023). Inspired by the success of combining multiple retrievers (Cormack et al., 2009), we adopt **fusion-based candidate generation**, aggregating diverse retrievers to reduce single-model bias and improve coverage. **LLM-as-a-Judge:** While LLMs show promise as relevance assessors (Rahmani et al., 2024), bias concerns remain (Soboroff, 2025). We therefore use LLMs primarily for filtering pooled candidates, identifying positives rather than generating them, and validate a subset with human annotators. **Index-Preserving Adaptation:** Standard symmetric fine-tuning (Karpukhin et al., 2020) incurs a prohibitive "Re-indexing Tax." Prior index-preserving approaches focus on pseudo-relevance feedback (Yu et al., 2021) or asymmetric tuning (Wang & Lyu, 2023). We propose **Query-Only Adaptation** and study it as an index-preserving deployment strategy in multi-tenant retrieval environments. **Parameter-Efficient Fine-Tuning (PEFT):** Methods such as Adapters (Houlsby et al., 2019), LoRA (Hu et al., 2022), embedding projections (Yoon et al., 2024), and partial layer unfreezing (Lee et al., 2023) have proven effective in retrieval (Litschko et al., 2022). Building on findings about intrinsic dimensionality (Aghajanyan et al., 2021), we show that applying PEFT to the query encoder achieves both strong performance and efficiency for multi-tenant serving.

3. Dataset Generation

Recent retrieval gains are driven by high-quality, domain-specific data, yet publicly available enterprise search datasets remain scarce. Existing benchmarks like BEIR (Thakur et al., 2021) do not capture the semi-structured, heterogeneous nature of enterprise data, including support tickets, issue trackers, and internal documentation. We seek to address this gap by providing a high-fidelity benchmark for enterprise-specific retrieval.

Manual annotation is costly and inherently low-recall, as annotators cannot exhaustively review large corpora, resulting in systematic false negatives. To address this, we introduce a scalable, automated dataset construction pipeline that maximizes candidate coverage through multi-stage retrieval while maintaining high precision via an LLM-as-a-judge.

3.1. Query Collection and Cleaning

We collected customer queries from production agent interactions as our source of real-world question data. However, raw customer queries often contain noise, including test queries, code snippets, and malformed inputs that are not legitimate natural language questions. To ensure dataset quality, we implemented a multi-stage filtering process: (1) **Length filtering**: Removing queries with word counts in the bottom and top 25% percentiles. (2) **Language detection**: Retaining only English queries. (3) **Deduplication**: Removing exact duplicate queries. (4) **Clustering-based diversity**: Selecting representative samples from clusters to ensure diversity and avoid semantic repetition. For further details on the above steps, please refer to Appendix A.2

3.2. Document Segmentation and Semantic Granularity

Enterprise documents are typically long and sparsely relevant to specific queries, which poses challenges for dense retrieval due to encoder token limits and the limited expressive capacity of fixed-size embeddings. Encoding entire documents often results in diluted representations that obscure fine-grained relevance. To address this, we collected all public documentation of DevRev and applied Recursive Character Splitting [2]. Documents are segmented into chunks of up to 500 characters with no overlap, maximizing semantic coverage while preserving precision. The recursive strategy favors natural structural boundaries (e.g., paragraphs, sentences, then spaces), ensuring each fragment is both embedding-friendly and semantically self-contained.

3.3. Multi-Retriever Annotation

To create high-quality query-document pairs, we employed an ensemble retrieval approach designed to maximize recall while maintaining precision.

Retrieval Ensemble: We applied an ensemble of seven diverse models: six dense retrievers (gemini-embedding-001 (Google, 2025), text-embedding-3-large (OpenAI, 2024), embed-english-v3 (Cohere, 2023), Qwen-3-Embedding-8B (Zhang et al., 2025), GTE-Qwen2-7B-Instruct (Li et al., 2023), SFR-Embedding-Mistral (Meng et al., 2024)) and one lexical retriever - BM25 (Robertson & Zaragoza, 2009), each returning the top 60 document chunks.

Union-based Aggregation: We computed the union of results from all retrievers to create a comprehensive candidate set of potentially relevant chunks. This ensures that chunks retrieved by any of the seven models are included in the candidate set, maximizing coverage and recall across different retrieval paradigms yielding ≥ 60 and ≤ 420 unique

[2]Documentation for LangChain's Recursive Text Splitter is available at the official LangChain documentation.

Table 1. Comparison of Recall@10 and NDCG@10 in retrieval effectiveness for baseline untrained model (Base) with Query-Document (QD), and Query-Only (Q) finetuning

Model	Variant	DevRev-Search		SciFact		FiQA	
		recall@10	ndcg@10	recall@10	ndcg@10	recall@10	ndcg@10
arctic-l-v2	Base	0.256	0.304	0.819	0.691	0.533	0.459
	QD	**0.314**	**0.362**	**0.869**	**0.748**	**0.578**	**0.505**
	Q	0.296	0.343	0.854	0.727	0.554	0.478
qwen3-4b	Base	0.219	0.264	0.903	0.769	0.649	0.571
	QD	0.325	**0.396**	0.949	**0.875**	**0.712**	**0.628**
	Q	**0.327**	0.370	**0.953**	0.830	0.688	0.602

candidate chunks per query across all models.

LLM-based Filtering: While ensembling improves recall, it also introduces false positives. To improve precision, we applied LLM-based filtering to the fused candidates using a prompt (Appendix A.4) that retains only chunks genuinely relevant to each query, removing lexically or semantically similar chunks lacking substantive answer content.

Quality Validation: To verify the reliability of our automated annotation process, we randomly sampled 10% queries and manually validated the final annotations to have a precision of 92%, which confirmed its effectiveness.

Retriever Contribution Analysis: Using the pipeline-generated chunks as ground truth, we evaluated individual retriever performance. A maximum recall of 82.48 for gemini-embedding-001 (Table 3, Appendix A.3) shows that single retrievers miss many relevant documents. We also performed leave-one-out ablations by aggregating candidates from the remaining six retrievers. As shown in Table 4 (Appendix A.3), recall varies from 93.25 to 97.13, indicating that each retriever contributes unique candidates and that combining dense and lexical retrieval improves coverage.

Our pipeline yields a high-quality dataset of 383 queries and 65.2K chunks, termed DevRev-Search, split into 291 training and 92 test queries. The dataset has high relevance density, averaging 13.61 golden chunks per query, with further details provided in Appendix A.1.

4. Query-Only Adaptation

Standard bi-encoder fine-tuning updates both the query (E_q) and document (E_d) encoders, but modifying E_d requires costly re-embedding and index reconstruction (e.g. HNSW (Malkov & Yashunin, 2020)) over possibly millions of documents, which is impractical in large-scale multi-tenant systems. To avoid this cost, we propose and study **Query-Only Adaptation**, which fine-tunes only the query encoder while keeping the document encoder and the index frozen.

We evaluate on three datasets with varying relevance densities: DevRev-Search (enterprise, high density; 13.6 relevant chunks/query), SciFact (scientific, low density; 1.1), and FiQA-2018 (financial, medium density; 2.6), emphasizing recall, precise evidence retrieval, and balanced precision–recall, respectively. We use `snowflake-arctic-embed-l-v2.0` (Yu et al., 2024) and `Qwen3-Embedding-4B` (Zhang et al., 2025) as backbone encoders.

Training uses InfoNCE loss with mined hard negatives to improve discrimination among similar documents. A fixed tuned temperature is used, and asynchronous ANCE training (Xiong et al., 2021) improves stability and performance. To address the limited size of DevRev-Search, we apply LLM-based augmentation, expanding the training set to 3014 queries. The open-sourced benchmark excludes these augmented queries, but they can be easily reproduced using the procedure in Appendix A.6.

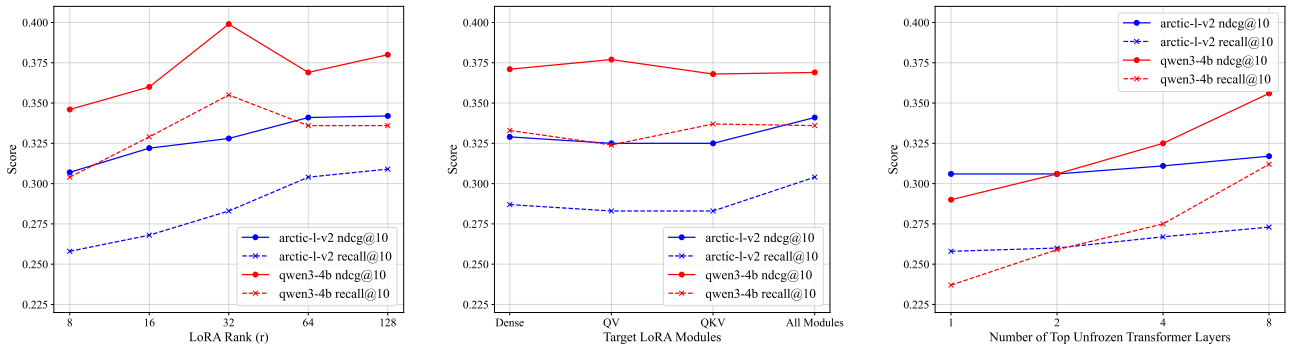
Our analysis focuses on: (1) **Query-Only vs. Query–Document Fine-Tuning**, measuring the performance impact of freezing the document index; and (2) **Parameter-Efficient Query Adaptation**, evaluating whether low-rank and parameter-efficient methods can match full fine-tuning while enabling cost-effective multi-tenant deployment.

4.1. Query-Only vs Query-Document Fine-Tuning

We evaluate whether asymmetric Query-Only fine-tuning (Q) can match symmetric Query-Document fine-tuning (QD). Table 1 summarizes results across both architectures and shows that freezing the document encoder incurs minimal performance loss. In all cases, Q substantially improves over the untrained baseline and remains competitive with the more expensive QD approach, achieving comparable retrieval metrics across domains and, for qwen3-4b on SciFact, slightly outperforming joint tuning in Recall@10.

Table 2. Comparison of retrieval effectiveness for various parameter-efficient methods with full fine-tuning for query-only adaptation (Here *LoRA and 8-Tr refer to the best LoRA configuration and fine-tuning only 8 transformer layers at the top of the base model, respectively)

Model	Type	DevRev-Search		SciFact		FiQA	
		recall@10	ndcg@10	recall@10	ndcg@10	recall@10	ndcg@10
arctic-l-v2	Linear	0.271	0.314	0.849	0.722	0.543	0.470
	FFN	0.281	0.318	0.880	0.743	0.534	0.464
	*LoRA	**0.309**	0.342	**0.915**	**0.827**	**0.555**	**0.478**
	8-Tr	0.273	0.317	0.849	0.718	0.547	0.476
	Full	0.296	**0.343**	0.854	0.727	0.554	**0.478**
qwen3-4b	Linear	0.331	0.358	0.935	0.815	0.666	0.583
	FFN	0.340	0.362	0.946	**0.861**	0.660	0.577
	*LoRA	**0.355**	**0.399**	0.950	0.844	**0.694**	**0.607**
	8-Tr	0.312	0.356	0.932	0.802	0.673	0.590
	Full	0.327	0.370	**0.953**	0.830	0.688	0.602



(a) *(b)* *(c)*

Figure 1. Ablation results for query-only adaptation for DevRev-Search across (a) varying LoRA ranks (with $\alpha = r$), (b) different combinations of target LoRA modules, and (c) varying number of top unfrozen transformer layers with the rest of the model frozen

4.2. Parameter Efficient Fine-Tuning

We evaluate whether Query-Only full fine-tuning performance can be achieved with PEFT methods that reduce trainable parameters for scalable multi-tenant adaptation. We test LoRA, a linear projection head, a 2-layer feed-forward network (FFN) on embeddings, and unfreezing top 8 transformer layers. Detailed comparison of trainable parameter counts is available in Table 6 (Appendix A.6).

Results in Table 2 show that best LoRA configuration consistently matches or outperforms full fine-tuning, likely due to implicit regularization from residual learning (Xin et al., 2024). Linear and FFN heads are also competitive, with FFN surpassing full fine-tuning on SciFact. In contrast, unfreezing top 8 layers underperforms despite using more parameters, highlighting LoRA's effectiveness.

LoRA Rank Ablations. We study sensitivity to the LoRA rank (r) with $\alpha = r$. On DevRev-Search (Fig. 1a), arctic-l-v2 improves monotonically and saturates at higher ranks, while qwen3-4b peaks at $r = 32$, likely due to overfitting at larger ranks. Results on SciFact and FiQA (Table 7, Appendix A.7) show a consistent pattern: $r \in [32, 64]$ provides the best trade-off between capacity and regularization. Although $r = 128$ can yield minor gains, performance largely saturates, making 32 or 64 the most efficient choices.

LoRA Target Module Ablations. We compare LoRA-based query-only adaptation across transformer sub-layer targets: dense, QV (Query-Value), QKV (Query-Key-Value), and all modules. Figure 1b shows DevRev-Search results, while SciFact and FiQA results appear in Table 8 (Appendix A.7). While adapting all modules performs best, tuning only dense layers offers a strong performance–efficiency trade-off, especially for qwen3-4b.

Transformer Layer Ablations. We freeze the base model and vary the number of unfrozen top transformer layers.

As shown in Fig. 1c, performance increases monotonically with more unfrozen layers on DevRev-Search, regardless of architecture. Results on the other datasets (Table 9, Appendix A.7) show the same trend.

5. Conclusion

In this paper we proposed an enterprise-focused retrieval benchmark constructed via a scalable pipeline that combines multi-retriever fusion and LLM-based filtering to generate high-quality annotated data. We also proposed and systematically studied query-only fine-tuning via PEFT as a practical approach to domain adaptation of dense retrieval models for multi-tenant search, wherein only the query encoder undergoes fine-tuning, and the document index remains frozen. Our results suggest that this approach provides strong performance without the re-indexing cost, while improving the quality–efficiency trade-off.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pp. 7319–7328, 2021.

Bonifacio, L. H., Abonizio, H. Q., Fadaee, M., and Nogueira, R. Inpars: Data augmentation for information retrieval using large language models. *CoRR*, abs/2202.05144, 2022. URL https://arxiv.org/abs/2202.05144.

Cohere. Introducing embed v3. https://cohere.com/blog/introducing-embed-v3, 2023. Accessed: 2026-01-06.

Cormack, G. V., Clarke, C. L., and Buettcher, S. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 758–759, 2009.

Dai, Z., Zhao, V. Y., Ma, J., Luan, Y., Ni, J., Lu, J., Bakalov, A., Guu, K., Hall, K. B., and Chang, M. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=gmL46YMpu2J.

Ester, M., Kriegel, H., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Simoudis, E., Han, J., and Fayyad, U. M. (eds.), *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pp. 226–231. AAAI Press, 1996. URL http://www.aaai.org/Library/KDD/1996/kdd96-037.php.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, D. M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pp. 249–256. JMLR.org, 2010. URL http://proceedings.mlr.press/v9/glorot10a.html.

Google. Embeddings. https://ai.google.dev/gemini-api/docs/embeddings, 2025. Accessed: 2026-01-06.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1026–1034. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.123. URL https://doi.org/10.1109/ICCV.2015.123.

Hendrycks, D. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL http://proceedings.mlr.press/v97/houlsby19a.html.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W. Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. Surgical fine-tuning improves adaptation to distribution shifts. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=APuPRxjHvZ.

Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., and Zhang, M. Towards general text embeddings with multi-stage contrastive learning. *CoRR*, abs/2308.03281, 2023. doi: 10.48550/ARXIV.2308.03281. URL https://doi.org/10.48550/arXiv.2308.03281.

Litschko, R., Vulic, I., and Glavas, G. Parameter-efficient neural reranking for cross-lingual and multilingual retrieval. In Calzolari, N., Huang, C., Kim, H., Pustejovsky, J., Wanner, L., Choi, K., Ryu, P., Chen, H., Donatelli, L., Ji, H., Kurohashi, S., Paggio, P., Xue, N., Kim, S., Hahm, Y., He, Z., Lee, T. K., Santus, E., Bond, F., and Na, S. (eds.), *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pp. 1071–1082. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.90.

Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M., and Balahur, A. Www'18 open challenge: Financial opinion mining and question answering. In Champin, P., Gandon, F., Lalmas, M., and Ipeirotis, P. G. (eds.), *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pp. 1941–1942. ACM, 2018. doi: 10.1145/3184558.3192301. URL https://doi.org/10.1145/3184558.3192301.

Malkov, Y. A. and Yashunin, D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473. URL https://doi.org/10.1109/TPAMI.2018.2889473.

Meng, R., Liu, Y., Joty, S. R., Xiong, C., Zhou, Y., and Yavuz, S. Sfr-embedding-mistral: Enhance text retrieval with transfer learning. Salesforce AI Research Blog, 2024. URL https://www.salesforce.com/blog/sfr-embedding/.

OpenAI. New embedding models and api updates. https://openai.com/index/new-embedding-models-and-api-updates, 2024. Accessed: 2026-01-06.

Rahmani, H. A., Craswell, N., Yilmaz, E., Mitra, B., and Campos, D. Llmjudge: Llms-as-judges for relevance assessments, 2024. URL https://arxiv.org/abs/2408.08896.

Robertson, S. E. and Zaragoza, H. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Soboroff, I. Don't use llms to make relevance judgments. *Inf. Retr. Res. J.*, 1(1):29–46, 2025. doi: 10.54195/IRRJ.19625. URL https://doi.org/10.54195/irrj.19625.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract-round2.html.

Wadden, D., Lo, K., Kuehl, B., Cohan, A., Beltagy, I., Wang, L. L., and Hajishirzi, H. Scifact-open: Towards open-domain scientific claim verification. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, volume EMNLP 2022 of *Findings of ACL*, pp. 4719–4734. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-EMNLP.347. URL https://doi.org/10.18653/v1/2022.findings-emnlp.347.

Wang, K., Thakur, N., Reimers, N., and Gurevych, I. GPL: generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In Carpuat, M., de Marneffe,

M., and Ruíz, I. V. M. (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 2345–2360. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.NAACL-MAIN.168. URL https://doi.org/10.18653/v1/2022.naacl-main.168.

Wang, Y. and Lyu, H. Query encoder distillation via embedding alignment is a strong baseline method to boost dense retriever online efficiency. In Moosavi, N. S., Gurevych, I., Hou, Y., Kim, G., Kim, Y. J., Schuster, T., and Agrawal, A. (eds.), *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing, SustaiNLP 2023, Toronto, Canada (Hybrid), July 13, 2023*, pp. 290–298. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.SUSTAINLP-1.23. URL https://doi.org/10.18653/v1/2023.sustainlp-1.23.

Xin, C., Lu, Y., Lin, H., Zhou, S., Zhu, H., Wang, W., Liu, Z., Han, X., and Sun, L. Beyond full fine-tuning: Harnessing the power of LoRA for multi-task instruction tuning. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N. (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 2307–2317, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.206/.

Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P. N., Ahmed, J., and Overwijk, A. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

Yoon, J., Chen, Y., Arik, S. Ö., and Pfister, T. Search-adaptor: Embedding customization for information retrieval. In Ku, L., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 12230–12247. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.661. URL https://doi.org/10.18653/v1/2024.acl-long.661.

Yu, H., Xiong, C., and Callan, J. Improving query representations for dense retrieval with pseudo relevance feedback. In Demartini, G., Zuccon, G., Culpepper, J. S., Huang, Z., and Tong, H. (eds.), *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pp. 3592–3596. ACM, 2021. doi: 10.1145/3459637.3482124. URL https://doi.org/10.1145/3459637.3482124.

Yu, P., Merrick, L., Nuti, G., and Campos, D. Arctic-embed 2.0: Multilingual retrieval without compromise. *CoRR*, abs/2412.04506, 2024. doi: 10.48550/ARXIV.2412.04506. URL https://doi.org/10.48550/arXiv.2412.04506.

Zhang, Y., Li, M., Long, D., Zhang, X., Lin, H., Yang, B., Xie, P., Yang, A., Liu, D., Lin, J., Huang, F., and Zhou, J. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176, 2025. doi: 10.48550/ARXIV.2506.05176. URL https://doi.org/10.48550/arXiv.2506.05176.

A. Appendix

A.1. Dataset Statistics

We partition the DevRev-Search dataset, consisting of 383 high-quality queries and 65.2K chunks, into a training set and an evaluation (test) set. The dataset consists of 291 training queries and 92 test queries. The training set exhibits a rich density of relevant documents, with an average of 13.61 golden chunks per query. Notably, the distribution of relevant documents is right-skewed, with a median of 6 and a high standard deviation ($\sigma = 21.41$). This variance reflects the diverse nature of enterprise search data, where some queries address specific technical identifiers with a single relevant source, while others address broad architectural topics with many relevant documentation fragments. To maintain the integrity of our public benchmark, we withhold the gold labels for the test queries to facilitate blind evaluation.

A.2. Query Cleaning

During exploratory analysis of the collected customer queries, we observed that queries at the extremes of the length distribution were often unsuitable. Very short queries (e.g., "Hello", "Help me please") typically lacked sufficient context to infer user intent and were frequently associated with test or placeholder inputs. Conversely, very long queries often consisted of pasted document excerpts, logs, or code snippets, reflecting attempts to process bulk content rather than genuine customer questions. To address this, we applied percentile-based length filtering, removing queries in the bottom and top 25% of the length distribution. Empirically, this range provided a balanced subset of queries that were more representative of natural user interactions and better suited for retrieval-based evaluation. We restricted the dataset to English-language queries as a deliberate design choice to construct a linguistically consistent benchmark. While this limits multilingual applicability, it reduces variability introduced by cross-lingual differences and enables more controlled evaluation. To prevent overrepresentation of repeated inputs, we removed exact duplicate queries. However, recognizing that lexical deduplication alone does not ensure semantic diversity, we further applied a clustering-based sampling strategy. Specifically, we embedded each query using the text-embedding-3-large model and performed clustering in the embedding space using DBSCAN (Ester et al., 1996). Queries were then sampled in a round-robin fashion across clusters until the desired dataset size was reached, which would allow for our annotation procedure under the cost constraints we were operating with. This approach ensured coverage of diverse semantic intents while avoiding redundancy.

A.3. Retriever Contributions

Analysis for retriever contributions is summarized in Tables 3 and 4.

Table 3. Recall@420 of Individual Retrievers on **DevRev-Search** dataset

Model	Recall
gemini-embedding-001	82.48
gte-Qwen2-7B-instruct	82.25
SFR-Embedding-Mistral	79.20
text-embedding-3-large	75.54
Qwen3-Embedding-8B	70.12
embed-english-v3	65.83
BM25	52.18

Table 4. Leave-one-out ablation study on **DevRev-Search** dataset

Model Combination	Recall
All $\setminus$ {gemini-embedding-001}	93.25
All $\setminus$ {gte-Qwen2-7B-instruct}	95.86
All $\setminus$ {SFR-Embedding-Mistral}	96.30
All $\setminus$ {text-embedding-3-large}	97.13
All $\setminus$ {Qwen3-Embedding-8B}	96.83
All $\setminus$ {embed-english-v3}	95.61
All $\setminus$ {BM25}	95.96

A.4. LLM-based Filtering Prompt

We used `gpt-5-2025-08-07` model under default decoding settings of temperature 1 and top-p 1. The prompt used for LLM-based filtering consists of annotation instructions, few-shot examples, and the target query-chunk pair. The full prompt is presented below.

A.4.1. System Prompt

```
Annotation Instructions
The focus is on whether an article chunk
    would help a support agent answer a
    query. Key instructions for annotators:

Focus on Problem in the query and
    Information in article chunk:
    Determine if the problem described in
    the query can be answered by the
    information present in article chunk.
    If article chunk's information would
    likely answer the query, then article
    chunk should be labeled as relevant.
    For example, if the article chunk
    explains how to use certain features
    in the app and the query is also
    asking how to use those features (even
    if in different words).
```

```
IMPORTANT - Beware of Superficial Word
    Overlap: Do not label an article chunk
    as relevant only because it shares
    some keywords with the query. Read the
    article chunk and query fully -
    article chunk and query might both
    mention a common term (like "login")
    but could be about different aspects
    of login (one about UI for the login
    page, another about authentication).
    Only consider lexical overlap
    meaningful if the article chunk
    contains information to answer the
    query (e.g. the query asks how to
    solve a specific login issue, and the
    article chunk contains information to
    solve that specific login issue).

{few_shot_examples}

Edge case: In the case article chunk
    contains only partial information
    required to answer the query, label it
    relevant only in the case when it
    answers the query substantially.
    Simple lexical overlap does not imply
    relevance. When in doubt, ask: "Would
    a support agent benefit from seeing
    the article chunk while answering the
    query?" If yes, label it similar; if
    not, or only minimally, then it's not
    relevant enough to help in the support
    workflow.
```

A.4.2. Few-Shot Examples

```
Examples of Relevant article chunk:

Example 1: Query: "Where can I find the
    DevRev API documentation?" and article
    chunk: "Resources to learn how to use
    DevRev APIs can be found at
    https://developer.devrev.ai/". The
    query is asking about where to find
    documentation on how to use DevRev
    APIs and the article chunk contains
    the information about the location
    where to find DevRev API
    documentation. The article chunk
    should be marked as relevant - it
    contains the information required to
    answer the query (even if words
    differ).

Example 2: Query: "What is a custom
    object?" and article chunk: "To create
    a custom object raise a support
    ticket. Custom objects are DevRev
    objects which can be customized". Even
    though the article chunk initially
    contains the information on how to
    create a custom object, it later also
    contains the information on what is a
    custom object which is what is asked
    in the query. Mark the article chunk
    relevant.

Examples of Non-Relevant article chunk:

Example 1: Query: "How to create a vista?"
    vs article chunk: "Vista is a list of
    DevRev objects" Both query and article
    chunk are about vistas and share the
    word "vista" but the information in
    article chunk is different from what
    query is asking about (query is asking
    how to create a vista, the article
    chunk is about what are vistas). This
    article chunk should be marked non
    relevant - information in the article
    chunk would not help answer the query.

Example 2: Query: "How to solve FORBIDDEN
    error when calling custom object API?"
    vs article chunk: "To solve
    BAD_REQUEST error when calling custom
    object API, look for DevRev custom
    object API documentation and fix your
    request structure". On the surface the
    article chunk looks relevant (same
    feature: custom object API). However,
    the error nature is different (one is
    a FORBIDDEN error, another is a
    BAD_REQUEST error). Unless further
    context in the article chunk reveals
    that both are the same errors, treat
    the article chunk as non relevant
    because the resolutions of both errors
    would differ (one might need more
    permissions, the other requires fixing
    the request).

Example 3: Query: Resource Center
    downloads tutorials API documentation
    vs article chunk: "Prerequisites\n*
    Send your first API request\n* Making
    a GET request\n* Next steps\n\nAPI
    Reference\n\nGetting
    started\n===============\n\nCopy
    page\n\nThe DevRev API is organized
    around REST. Our API has predictable
    resource-oriented URLs, accepts". On
    the surface the article chunk appears
    to be answering the query because it
    has links to the documentation but the
    problem is that it is part of start of
    a webpage so only has relative links
    and not actual content or complete URL.
```

A.4.3. User Prompt Template

```
Query: {query}
Article Chunk: {candidate}
```

A.5. Motivation for the Multi-Tenant Deployment Setting

In our multi-tenant scenario, each tenant maintains its own document corpus and independent document index. No index is shared across tenants, ensuring strict privacy and isolation. Multi-tenancy is achieved through a shared base dense retriever combined with tenant-specific query adapters implemented as lightweight LoRA modules, or other PEFT adapters. Each tenant trains its adapter using only its own queries, and at serving time, incoming queries are routed to the appropriate tenant so that the corresponding adapter is dynamically activated while the base model remains shared. This design keeps both indices and adapters fully isolated while enabling efficient multi-tenant retrieval.

This approach is motivated by practical considerations. Classical fine-tuning would require updating both query and document encoders and re-indexing each tenant's corpus—an expensive and operationally heavy process. By adapting only the query side, document embeddings and indices remain fixed, avoiding re-indexing. In our experiments, we simulate heterogeneous tenants using three diverse domains (Enterprise, Scientific, Financial) with different relevance distributions. Tenant-specific adapters are lightweight and inexpensive to store and manage, demonstrating the feasibility of this deployment approach in practice.

A.6. Experimental Setup

All our training optimizes InfoNCE loss using AdamW optimizer, with 16 mined hard negatives per query, with 8 negatives randomly sampled from them each time. A learning rate of 5×10^{-6} is used for all runs with a cosine scheduler and no warm-up. For all the experiments, we use a uniform temperature of 0.1, which was found to work the best.

Additionally, we noticed that simply using fixed mined hard negatives leads to cases of representation collapse wherein InfoNCE loss keeps improving, but training metrics rapidly deteriorate after a point, especially in datasets like FiQA and DevRev-Search. Upon further analysis, we noticed that this happens due to evolving hard negatives, where over the course of training, old hard negatives become 'simple' and new hard negatives replace them. To counteract this, we adopt the asynchronous hard negative refresh technique proposed by ANCE (Xiong et al., 2021) to periodically update the hard negatives in InfoNCE loss at every 200 steps. This proved to improve the performance in most cases, in addition to much more stable training as shown in Fig 2. Quantitative comparison of results obtained with and without this technique is shown in Table 5

Training Set Augmentation: For augmenting the training set of DevRev-Search, we used a carefully prompted LLM to take in an individual query as well as their ground truth chunks, and generate a set of new queries, where each of the generated query addresses concepts/topics/questions that are subsets of the concepts/topics/questions addressed in the original query from which it was created. This achieves three goals : (1) the ground truth chunks pertaining to the generated query would be a subset of the ground truth chunks of the original query, (2) the generated queries follow the same 'distribution' as the original queries, (3) the generated queries are strictly semantically distinct from the original query so as to ensure no data duplication. Applying this strategy to the original 291 train queries resulted in a total of 3014 queries. Following this, for each of the generated queries, we filter out the irrelevant chunks from the ground truth chunks of the original query using an LLM, which we then take as the ground truth chunks for the generated query. This resulted in a diverse augmented set that pertained to the same 'distribution' and, at the same time, mitigated risks that came from a small training set.

The feed-forward network (FFN) used in our PEFT experiments consists of GELU (Hendrycks, 2016) activations and 2 hidden layers, each with the same number of nodes as the base model's embedding dimension. Also, for both linear and FFN heads, we find that zeroing biases and initializing weight matrices to the identity perform consistently better than standard initializations such as Xavier (Glorot & Bengio, 2010) or Kaiming (He et al., 2015).

A.7. Ablation Results

Tables 7, 8, and 9 present the ablation results across different LoRA ranks, different LoRA target modules, and different numbers of unfrozen transformer layers of the dense model, for Query-Only fine-tuning, respectively.

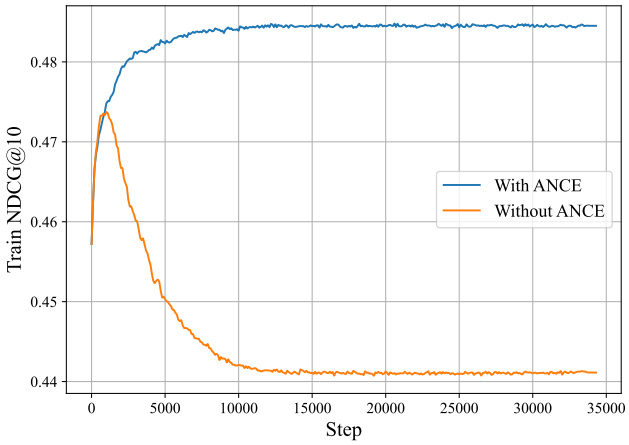


Figure 2. Training NDCG@10 vs training step for Query-Only adaptation of arctic-l-v2 (full model)

Table 5. Retrieval effectiveness with and without ANCE for full Query-Only adaptation

Model		DevRev-Search		SciFact		FiQA	
		recall@10	ndcg@10	recall@10	ndcg@10	recall@10	ndcg@10
arctic-l-v2	Standard	0.291	0.331	**0.860**	**0.731**	0.543	0.473
	+ANCE	**0.296**	**0.343**	0.854	0.727	**0.554**	**0.478**
qwen3-4b	Standard	**0.333**	**0.373**	0.941	0.813	0.679	0.593
	+ANCE	0.327	0.370	**0.953**	**0.830**	**0.688**	**0.602**

Table 6. Parameter efficiency comparison for Query-Only adaptation across LoRA ranks, unfrozen layers, and head-only training.

Model	Configuration	Trainable Params	Total Params	% Full
arctic-l-v2	Full Model	566,705,152	566,705,152	100.00
	LoRA $r = 8$	3,538,944	570,244,096	0.62
	LoRA $r = 16$	7,077,888	573,783,040	1.25
	LoRA $r = 32$	14,155,776	580,860,928	2.50
	LoRA $r = 64$	28,311,552	595,016,704	4.99
	LoRA $r = 128$	56,623,104	623,328,256	9.99
	Linear Head (Frozen Base)	1,049,600	567,754,752	0.19
	MLP Head (Frozen Base)	3,148,800	569,853,952	0.56
	1 Unfrozen Layer	12,596,224	566,705,152	2.22
	2 Unfrozen Layers	25,192,448	566,705,152	4.45
	4 Unfrozen Layers	50,384,896	566,705,152	8.89
	8 Unfrozen Layers	100,769,792	566,705,152	17.78
qwen3-4b	Full Model	4,021,774,336	4,021,774,336	100.00
	LoRA $r = 8$	16,515,072	4,038,289,408	0.41
	LoRA $r = 16$	33,030,144	4,054,804,480	0.82
	LoRA $r = 32$	66,060,288	4,087,834,624	1.64
	LoRA $r = 64$	132,120,576	4,153,894,912	3.29
	LoRA $r = 128$	264,241,152	4,286,015,488	6.57
	Linear Head (Frozen Base)	6,556,160	4,028,330,496	0.16
	MLP Head (Frozen Base)	19,668,480	4,041,442,816	0.49
	1 Unfrozen Layer	100,933,376	4,021,774,336	2.51
	2 Unfrozen Layers	201,864,192	4,021,774,336	5.02
	4 Unfrozen Layers	403,725,824	4,021,774,336	10.04
	8 Unfrozen Layers	807,449,088	4,021,774,336	20.08

Table 7. Retrieval effectiveness variation across different LoRA ranks ($r = \alpha$ for all configurations)

Model	LoRA Rank	DevRev-Search		SciFact		FiQA	
		recall@10	ndcg@10	recall@10	ndcg@10	recall@10	ndcg@10
arctic-l-v2	8	0.258	0.307	0.867	0.738	0.540	0.469
	16	0.268	0.322	0.880	0.748	0.548	0.474
	32	0.283	0.328	0.905	0.769	0.554	**0.478**
	64	0.304	0.341	0.914	0.807	**0.555**	**0.478**
	128	**0.309**	**0.342**	**0.915**	**0.827**	**0.555**	**0.478**
qwen3-4b	8	0.304	0.346	0.948	0.826	0.685	0.603
	16	0.329	0.360	0.948	0.832	0.686	0.605
	32	**0.355**	**0.399**	0.950	**0.844**	0.690	**0.607**
	64	0.336	0.369	0.950	0.816	**0.694**	0.607
	128	0.336	0.380	**0.957**	0.831	0.682	0.601

Table 8. Retrieval effectiveness variation across different target modules for LoRA $r = 64$ and $\alpha = 64$

Model	Layers	DevRev-Search		SciFact		FiQA	
		recall@10	ndcg@10	recall@10	ndcg@10	recall@10	ndcg@10
arctic-l-v2	Dense	0.287	0.329	0.854	0.726	0.551	0.477
	QV	0.283	0.325	0.854	0.726	0.554	0.476
	QKV	0.283	0.325	0.854	0.726	0.554	0.477
	All	**0.304**	**0.341**	**0.914**	**0.807**	**0.555**	**0.478**
qwen3-4b	Dense	0.333	0.371	**0.950**	**0.818**	0.693	**0.607**
	QV	0.324	**0.377**	0.923	0.793	0.688	0.605
	QKV	**0.337**	0.368	0.930	0.796	0.689	0.606
	All	0.336	0.369	**0.950**	0.816	**0.694**	**0.607**

Table 9. Retrieval effectiveness variation across different numbers of unfrozen transformer layers at the top of the base model, with the rest of the layers frozen

Model	Layers	DevRev-Search		SciFact		FiQA	
		recall@10	ndcg@10	recall@10	ndcg@10	recall@10	ndcg@10
arctic-l-v2	1	0.258	0.306	0.827	0.700	0.534	0.462
	2	0.260	0.306	0.835	0.709	0.536	0.464
	4	0.267	0.311	0.835	0.712	0.539	0.471
	8	**0.273**	**0.317**	**0.849**	**0.718**	**0.547**	**0.476**
qwen3-4b	1	0.237	0.290	0.903	0.778	0.661	0.576
	2	0.259	0.306	0.908	0.783	0.664	0.579
	4	0.275	0.325	0.912	0.788	0.667	0.583
	8	**0.312**	**0.356**	**0.932**	**0.802**	**0.673**	**0.590**