

ENTERPRISE RAG RETRIEVAL

Domain-adapting a retrieval embedding model with LoRA

A case study on **Corporate GPT**

AUTHOR

Neurologic AI

PLATFORM

Corporate GPT

BASE MODEL

Owen3-Embedding-8B

ABSTRACT

The answer quality of a retrieval-augmented generation (RAG) system is upper-bounded by its retrieval step: a generator can only ground its output in the passages it is given. In enterprise RAG, that step is driven by an embedding model that is typically a strong general-purpose model with no exposure to a customer's domain vocabulary or query distribution. We study whether lightweight domain adaptation of this embedding model improves retrieval, and therefore answer grounding, in Corporate GPT, a production enterprise RAG platform. We adapt

Qwen3-Embedding-8B using Low-Rank Adaptation (LoRA) with an InfoNCE contrastive objective, trained on a fully synthetic, label-free dataset built by LLM query generation and hard-negative mining over the domain corpus. We evaluate the result two independent ways: (i) an offline retrieval-ranking test on a held-out document, and (ii) an end-to-end A/B test on the live pipeline, scored by an automated RAG-evaluation framework with an independent LLM judge, changing only the embedder. Both evaluations agree. Offline, Recall@1 improves by +0.098 and MRR@10 by +0.081. On the live pipeline, every retrieval metric improves (context utilization by +0.108 and context recall by +0.048) and answer faithfulness rises by +0.043, while answer relevancy remains flat, as expected for a retrieval-only change. These gains were obtained on a single GPU at roughly 44% of one training epoch, indicating substantial headroom. We conclude that LoRA-based domain adaptation of the embedder is a cheap, reversible, and effective lever for enterprise RAG.

+0.098

Recall@1, offline

+0.081

MRR@10, offline

+0.108

Context utilization,
live pipeline

+0.043

Answer faithfulness,
live pipeline

01

Introduction

Retrieval-augmented generation couples a retrieval module with a language model, supplying the model with knowledge drawn from an external corpus at inference time and reducing hallucination relative to closed-book generation. A useful mental model is an open-book exam: the system first locates relevant passages (retrieval) and then composes an answer from them (generation). If retrieval surfaces the wrong passages, no amount of generation quality can recover the correct answer. Retrieval is therefore the binding constraint on end-to-end quality.

RETRIEVAL

Locate relevant passages

The binding constraint on end-to-end quality



GENERATION

Compose an answer from them

Can only ground its output in the passages it is given

In Corporate GPT (an enterprise RAG platform that answers employees' natural-language questions from an organization's own documents, with citations and access control) the retrieval step is implemented by a dense embedding model that maps both queries and document chunks into a shared vector space and returns nearest neighbours. Off-the-shelf embedding models are strong generalists trained on broad corpora, but they have not observed a given deployment's domain vocabulary, phrasing, or the mapping between its questions and its passages. Domain adaptation (nudging the embedding space toward the target domain) is a well-established route to improved retrieval that leaves the remainder of the stack unchanged.

Full fine-tuning of an 8-billion-parameter embedding model is costly and risks degrading the base model's general capability. We instead use Low-Rank Adaptation (LoRA), which trains a small set of adapter parameters while freezing the base model, yielding cheap, reversible, and low-risk experiments. We test a single hypothesis: adapting the embedding model to the domain improves retrieval, and thereby answer grounding, without modifying the generator.

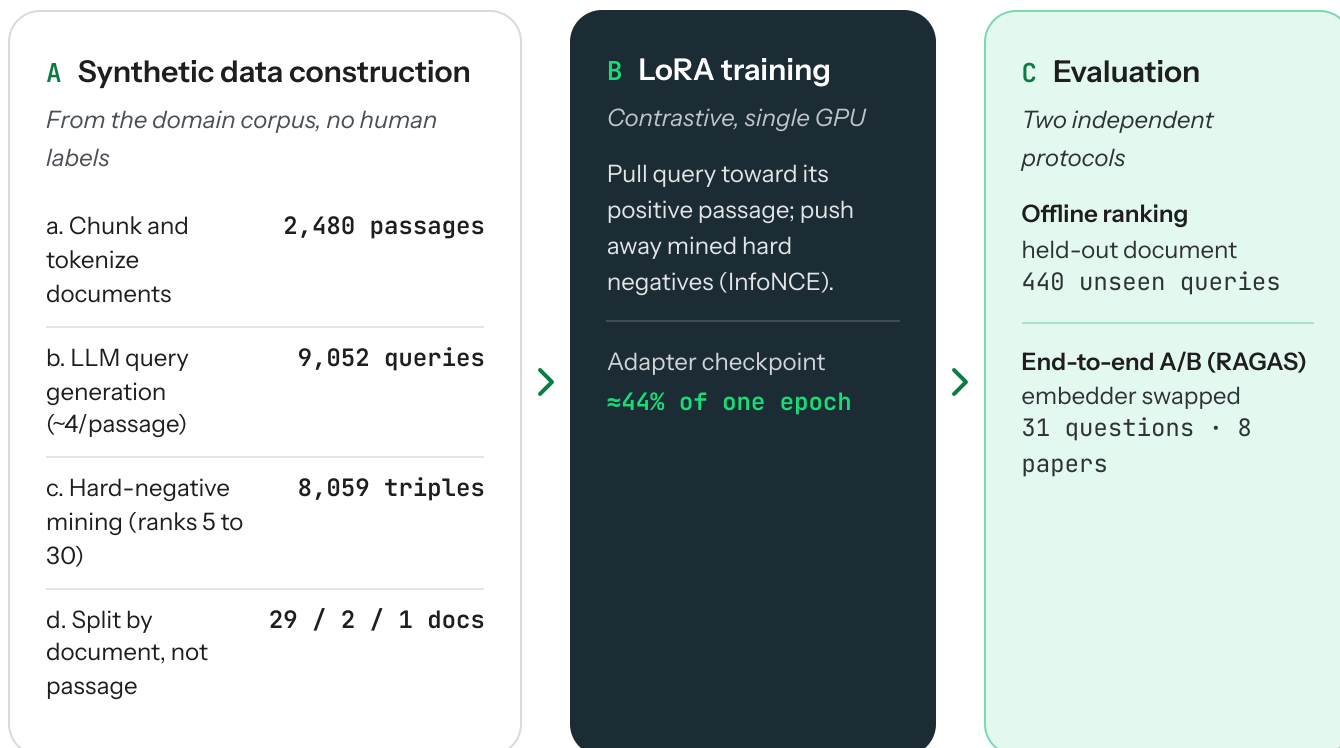


Figure 1. The self-contained pipeline: synthetic label-free data construction (A), LoRA contrastive adaptation (B), and two independent evaluations (C). Counts correspond to the 32-document source run.

Stage	Output	Count
Corpus	Passages after chunking (384-token target, 64 overlap)	2,480
Query generation	Synthetic questions (≈ 4 per passage)	9,052
Hard-negative mining	(query, positive, hard-negatives) triples	8,059
Document split	Train / validation / test, by document	29/2/1

Table 1. Synthetic training-data statistics (32-document source run).

Our contributions are (i) a fully synthetic, label-free training pipeline that domain-adapts an 8B embedding model with LoRA; (ii) a two-pronged evaluation that isolates the embedder, comprising an offline retrieval-ranking test and an end-to-end A/B test scored by an independent judge; and (iii) empirical evidence of consistent retrieval and grounding gains obtained at under half a training epoch on a single GPU, with two independent evaluations in agreement.

i

A fully synthetic, **label-free training pipeline** that domain-adapts an 8B embedding model with LoRA.

ii

A **two-pronged evaluation** that isolates the embedder.

iii

Empirical evidence of **consistent retrieval and grounding gains** at under half a training epoch on a single GPU.

 02

Method

2.1 Synthetic training-data generation

We assume no hand-labelled (query, passage) pairs and construct them from the corpus. Source documents are chunked and tokenized; an instruction-tuned model then generates candidate questions for each chunk; a mining stage embeds the corpus and retrieves near-neighbours to obtain hard negatives; and the resulting triples are split for training and evaluation. Figure 1 summarizes the pipeline and Table 1 reports the corresponding dataset statistics for the 32-document source run used throughout.

Two design decisions are central to the validity of the results. First, we split by whole document rather than by passage: near-duplicate chunks from a single document would otherwise appear in both the training and test partitions and inflate scores, so partitioning at the document level guarantees that evaluation material is unseen during training. Second, hard-negative mining targets middle ranks (5 to 30) rather than the top neighbours: the closest neighbours of a query are frequently themselves relevant, and treating them as negatives injects label noise; we therefore skip the top ranks, apply a similarity-margin filter, and forbid a negative from originating in the positive's own document.

DESIGN DECISION 1

Split by whole document

29 train

2 validation 1 test

Evaluation material is unseen during training.

DESIGN DECISION 2

Mine hard negatives from middle ranks

1 to 4

Ranks 5 to 30

Top ranks

skipped

Margin filter, not from the positive's document

The closest neighbours are often themselves relevant.

2.2 Model and adaptation

The base model is Qwen3-Embedding-8B (4096-dimensional embeddings, last-token pooling). We attach LoRA adapters (feature-extraction mode) of rank 16 with scaling factor $\alpha = 32$ and dropout 0.05 to all seven projection matrices (`q` , `k` , `v` , `o` , `gate` , `up` , `down`). The base weights are frozen throughout, so the original model is recoverable exactly, and serving can switch between the base and adapted models via a single configuration flag.

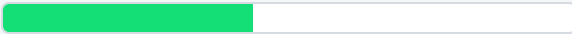
Component	Setting
Base model	Qwen3-Embedding-8B, 4096-dim, last-token pooling
Adaptation	LoRA (feature-extraction), rank 16, $\alpha = 32$, dropout 0.05
Target modules	all seven projections (<code>q</code> , <code>k</code> , <code>v</code> , <code>o</code> , <code>gate</code> , <code>up</code> , <code>down</code>)
Objective	InfoNCE, temperature 0.05, 6 hard negatives / query
Optimization	LR 1×10^{-4} , cosine schedule, 5% warmup
Batch	$4/\text{device} \times 2 \text{ grad-accum} = \text{effective } 8$
Precision	bf16 with gradient checkpointing
Sequence lengths	query 128 tokens, document 512 tokens
Seed	17 (single run)
Hardware	1× NVIDIA L40S (48 GB), $\approx 6.5 \text{ h/epoch}$
Checkpoint reported	step 800 $\approx 44\%$ of one epoch

Table 2. Adapter-training configuration.

2.3 Training objective and setup

We train with the InfoNCE contrastive loss at temperature 0.05, using six hard negatives per query. Optimization uses a learning rate of 1×10^{-4} with a cosine schedule and 5% warmup, an effective batch size of 8 (4 per device with 2-step gradient accumulation), bf16 precision, and gradient checkpointing. Query and document sequence lengths are capped at 128 and 512 tokens respectively; the random seed is 17. Training ran on a single NVIDIA L40S (48 GB): one full epoch is 1,825 steps at roughly 14 s/step (≈ 6.5 h). We report the checkpoint at step 800 ($\approx 44\%$ of one epoch); peak memory was ≈ 24 GB of 46 GB, leaving substantial room to increase batch size in future runs. Table 2 lists the complete configuration.

TRAINING PROGRESS 800 / 1,825 steps



Reported checkpoint at $\approx 44\%$ of one epoch

PEAK MEMORY ≈ 24 / 46 GB



Substantial room to increase batch size

03

Experimental setup

We measure impact two independent ways. A single metric can mislead; agreement between two protocols that use different test sets and different measurement mechanisms is stronger evidence than any one figure.

PROTOCOL 1

Offline retrieval

Test set 440 queries, one held-out document

Mechanism Rank all 2,480 passages

Language model None in the loop

PROTOCOL 2

End-to-end A/B on the live pipeline

Test set 31 questions, 8 unseen papers

Mechanism Only the embedder changes

Judge Qwen3.6-35B-FP8, RAGAS

3.1 Offline retrieval

The first protocol isolates the embedder with no language model in the loop. We take 440 queries associated with a single document held out of training entirely and rank all 2,480 corpus passages for each query, under the base and adapted models. We report Recall@{1, 5, 10}, nDCG@10, and MRR@10. This protocol measures ranking quality directly, without confounding from the generator.

3.2 End-to-end A/B on the live pipeline

The second protocol measures what a user experiences. We run the complete retrieve-then-generate pipeline and change only the embedder: the vector index is rebuilt for each model, while every downstream component (including the generator that composes answers) is held fixed. Because the generator is identical across arms, any movement in pipeline metrics is attributable to retrieval rather than to a more capable generator; this also implies that a purely generation-side metric (answer relevancy) should remain flat, which serves as a sanity check that the comparison isolates the intended variable. The test set comprises 31 questions spanning 8 papers that were never used in training: a generalization test broader than the offline probe. Answers are graded by an independent model (Qwen3.6-35B-FP8) under the RAGAS framework, so scoring does not depend on the system judging itself.

— 04

Results

4.1 Offline retrieval

Across all 440 held-out queries, the adapted model ranks the correct passage better on every measure (Figure 2, Table 3). The largest gain is at Recall@1 (+0.098): the model places the single correct passage first substantially more often, which is precisely the behaviour a retrieval-focused adaptation should induce. MRR@10 rises by +0.081 and nDCG@10 by +0.067, indicating that improvements are concentrated near the top of the ranking, where they matter most for downstream generation.

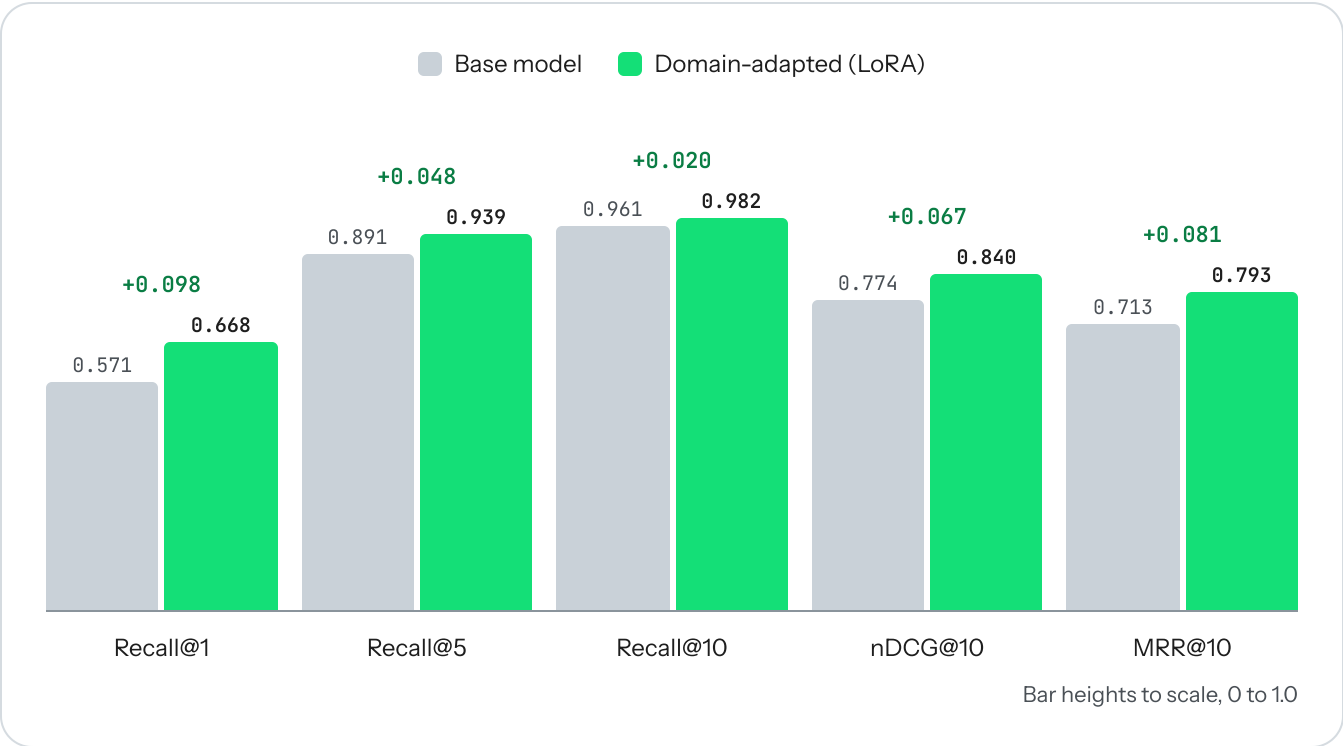


Figure 2. Offline retrieval on 440 held-out queries from an unseen document, ranked against all 2,480 passages. Higher is better; deltas (adapted – base) are annotated above each pair.

Metric	Base	Adapted	Δ
Recall@1	0.5705	0.6682	+0.098
Recall@5	0.8909	0.9386	+0.048
Recall@10	0.9614	0.9818	+0.020
nDCG@10	0.7736	0.8402	+0.067
MRR@10	0.7125	0.7932	+0.081

Table 3. Offline retrieval ranking (base vs. domain-adapted).

4.2 End-to-end pipeline

On the full retrieve-then-answer system, every retrieval metric improves (Figure 3, Table 4). The headline is context utilization (+0.108): the adapted model ranks genuinely useful passages near the top far more often. Context precision (+0.035) and context recall (+0.048) rise as well. Answer faithfulness (a generation-side measure of whether the answer is supported by the retrieved passages) increases by +0.043; with better source material

available, the generator invents less and adheres more closely to the documents. This grounding gain is a downstream consequence of improved retrieval rather than any change to the generator.

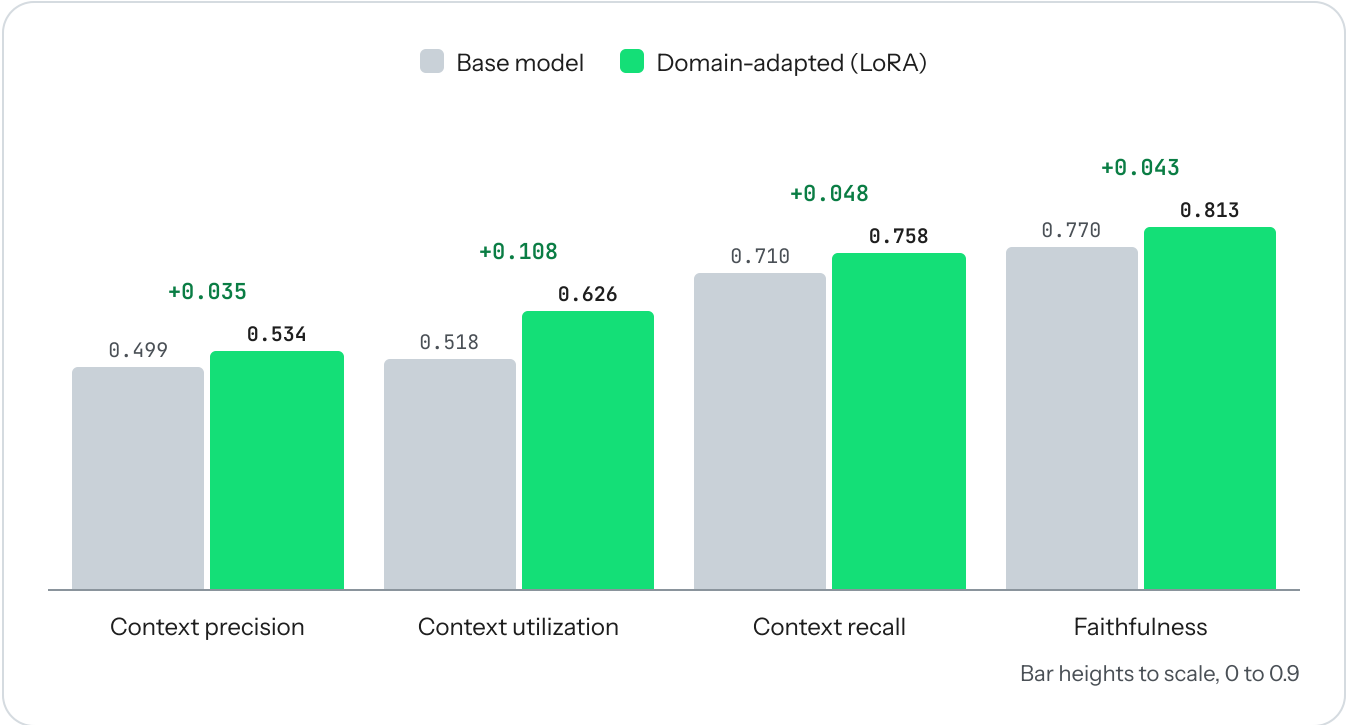


Figure 3. End-to-end A/B on 31 questions over 8 unseen papers, graded by an independent judge under RAGAS. The four metrics an embedder change is expected to move all rise; the two generation-side controls stay near-flat.

Metric	Base	Adapted	Δ
Context precision	0.4991	0.5342	+0.035
Context utilization	0.5181	0.6264	+0.108
Context recall	0.7097	0.7581	+0.048
Faithfulness	0.7697	0.8128	+0.043
Answer relevancy	0.8124	0.8048	-0.008†
Context entity recall	0.3567	0.3756	+0.019‡

Table 4. End-to-end pipeline (RAGAS), independently graded.

† Answer relevancy reflects how the answer is written and depends on the generator, not the embedder; a near-flat result is the expected outcome for a retrieval-only change, and serves as a positive control.

‡ Context entity recall is computed over the questions scored in both runs; with one question unscored per run, its coverage is marginally lower than the other five metrics.

4.3 Agreement between the two evaluations

The two protocols agree in direction and magnitude of the effect. Offline retrieval (Recall@1 +0.098) and the live pipeline (context utilization +0.108, context recall +0.048) point the same way from different test sets and different measurement mechanisms. Independent agreement is the strongest available signal that the improvement is real rather than an artifact of a single metric or a single document.

OFFLINE RETRIEVAL

Recall@1 **+0.098**

AGREE

LIVE PIPELINE

Context utilization **+0.108**, context recall **+0.048**

— 05

Discussion

5.1 Interpretation

The pattern of results is consistent with an embedder that has learned the domain's query-to-passage mapping. Gains are largest at the top of the ranking (Recall@1, context utilization), which is where retrieval most affects generation, since the generator conditions on the highest-ranked passages. The faithfulness increase, obtained without any change to the generator, indicates that better retrieval propagates into better-grounded answers: the intended end-to-end effect. The flat answer-relevancy result is a positive control: a retrieval-only change should not alter how answers are phrased, and it does not.

5.2 Practical implications

Because the base model is frozen and the adapter is a small artifact, promotion to production is a reversible, low-risk change: the serving path toggles between base and adapted models with a single flag, and reverting is exact. The method is also inexpensive (a single GPU, hours of training, no human labels) and therefore repeatable as a per-corpus onboarding procedure rather than a one-off experiment. Natural next steps include training for at least one full epoch with a larger batch and re-running the identical A/B; incorporating

anonymized production queries alongside the synthetic set; and scaling hard-negative mining by reusing the vectors already stored in the production index, avoiding re-embedding and representation drift.

NEXT

Train for at least one full epoch with a larger batch and re-run the identical A/B.

NEXT

Incorporate anonymized production queries alongside the synthetic set.

NEXT

Scale hard-negative mining by reusing the vectors already stored in the production index.

— 06

Conclusion

Domain-adapting the embedding model with LoRA measurably improved retrieval (and, in turn, answer grounding) in a production enterprise RAG system, as confirmed by two independent evaluations that agree. The gains (offline Recall@1 +0.098; live context utilization +0.108 and faithfulness +0.043) were achieved on a single GPU, with fully synthetic training data, at under half an epoch of training. The method is cheap, reversible, and has clear headroom above the figures reported here. It is a strong candidate for production and a repeatable template for adapting Corporate GPT to any customer's documents.

THE CONCLUSION

LoRA-based domain adaptation of the embedder is a **cheap, reversible, and effective lever for enterprise RAG.**