



[Home](#) > How We Built a World-Class Reranker for Fin

How We Built a World-Class Reranker for Fin

We built our own reranker that outperforms Cohere Rerank v3.5, an industry-leading commercial solution. This improved our answer quality, reduced reranking costs by 80%, and gained more flexibility to evolve our system.



Ramil Yarullin

2025.09.11



Contents

Fin's RAG Workflow

Why Build Our Own Reranker?

Fin-cx-reranker: Our Custom Solution

Training Details

[Evaluation](#)[What's Next](#)

At Intercom, Fin AI Agent uses retrieval-augmented generation (RAG) to deliver fast, accurate answers to customer support questions.

In this setup, a reranker plays a crucial role: after retrieving potential answers from our knowledge base, the reranker reorders them by relevance to help Fin choose the best content to include in its reply.

We built our own reranker that outperforms Cohere Rerank v3.5, an industry-leading commercial solution. This improved our answer quality, reduced reranking costs by 80%, and gained more flexibility to evolve our system.

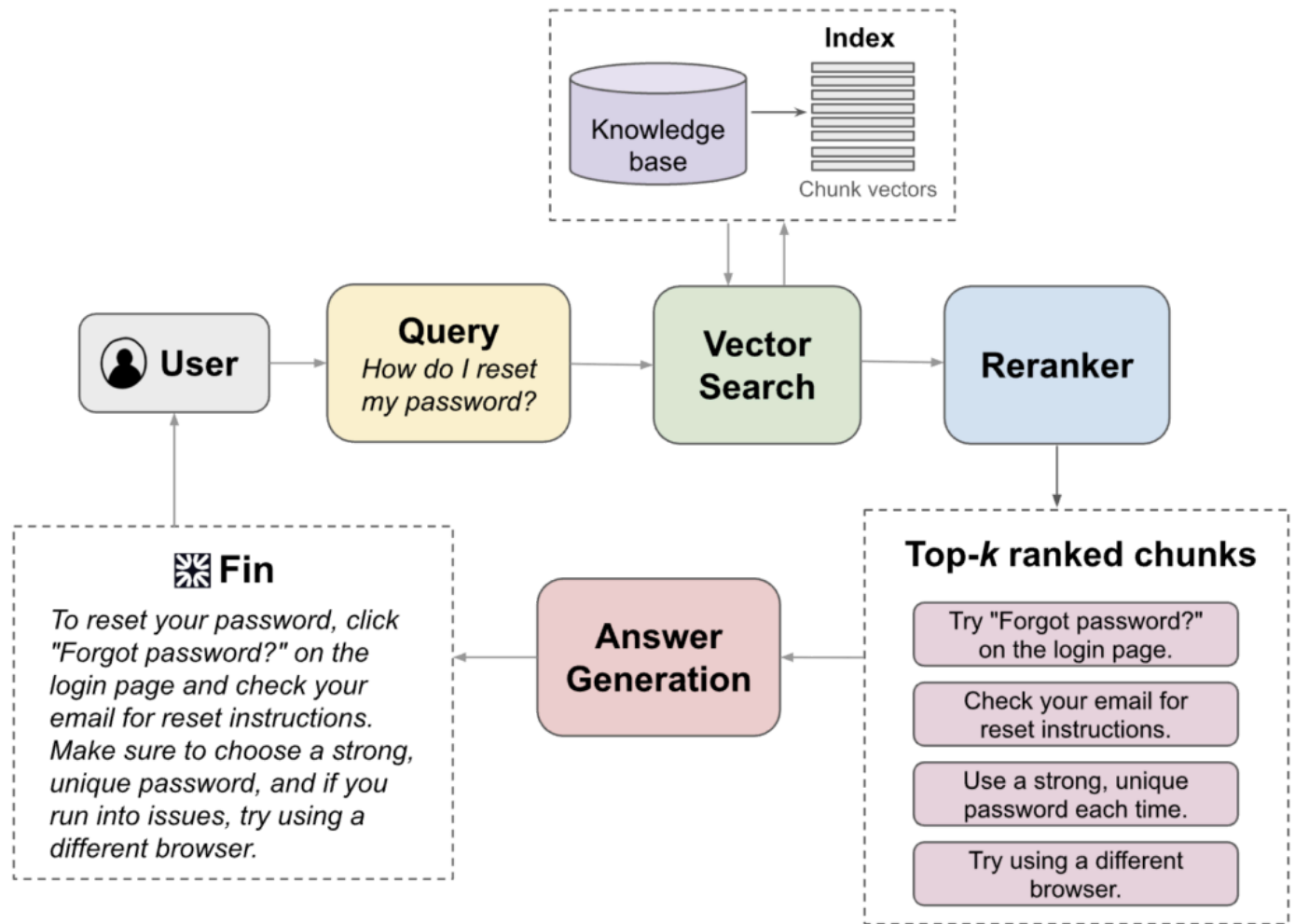
Fin's RAG Workflow

Here's how Fin uses RAG at a high level.

When someone asks Fin for help, Fin starts by summarizing the conversation into a short, focused query, like *"How do I reset my password?"* or *"Where can I find my invoices?"*. This query is used to search the knowledge base, where all help articles and snippets are pre-processed into vector embeddings for efficient retrieval.

Fin compares the query embedding to these vectors to find the closest matches. It then takes the top $K = 40$ candidates and re-ranks them using a specialized reranker model. Initial vector retrieval is fast, but can miss nuances, so the reranker uses deeper context understanding to reorder the passages by relevance.

Finally, a context budget filter selects the top-ranked passages, and Fin uses these to craft a clear, accurate answer for the user in real-time.



Why Build Our Own Reranker?

Previously, we relied on Cohere Rerank-v3.5, a commercial reranker offering high-quality results but incurring substantial costs. Previously tested open-source models (BGE-large and BGE-m3) couldn't achieve required performance levels, and using LLM-based reranker caused latency issues.

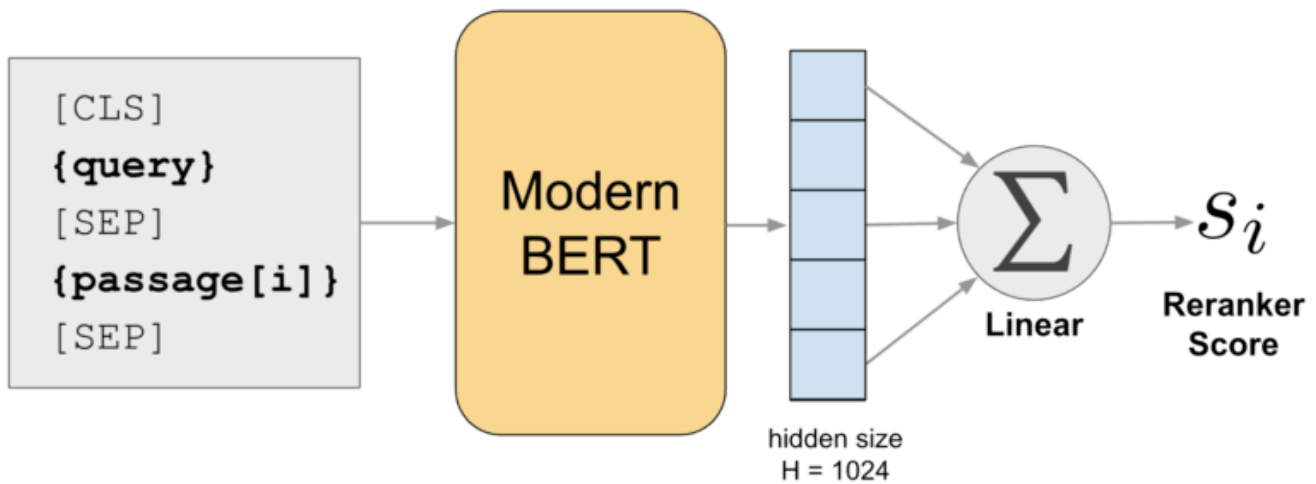
To address these challenges, we decided to develop our own reranker tailored specifically to the domain of English customer support. Our objectives were clear and ambitious: match or exceed Cohere's quality, run efficiently on standard GPUs, and reduce vendor dependency.

Fin-cx-reranker: Our Custom Solution

Our custom reranker uses ModernBERT-large (2024) as a component. This is a state-of-the-art encoder-only transformer designed specifically for retrieval and classification tasks. ModernBERT supports an 8,192-token context window (vs. 512 in vanilla BERT), employs rotary/relative positional encodings, GeGLU activations, efficient attention, and

was trained on ~2T tokens. It consistently surpasses encoders like BERT, RoBERTa, and DeBERTaV3 across benchmarks such as BEIR and GLUE by 3-8pp.

For scoring candidate passages, we concatenate each query and passage pair as **[CLS] {query} [SEP] {passage[i]} [SEP]** and feed this into ModernBERT. We then apply mean pooling across all token embeddings (excluding padding) to obtain a single vector. This vector passes through a linear layer, producing a final relevance score used for ranking.



Training Details

We trained the reranker on 400,000 real Fin queries, each with $K = 40$ candidate passages, 16M pairs in total. Labels were provided by an LLM-based pointwise reranker, giving us high-quality training signals.

Our implementation uses Hugging Face Transformers. To optimize ranking, we employ a RankNet loss. The teacher LLM first sorts the K passages by relevance, assigning each passage a rank r_i where a lower number means higher relevance (e.g., $r_i = 1$ means top-ranked). The model then produces a score s_i for each passage.

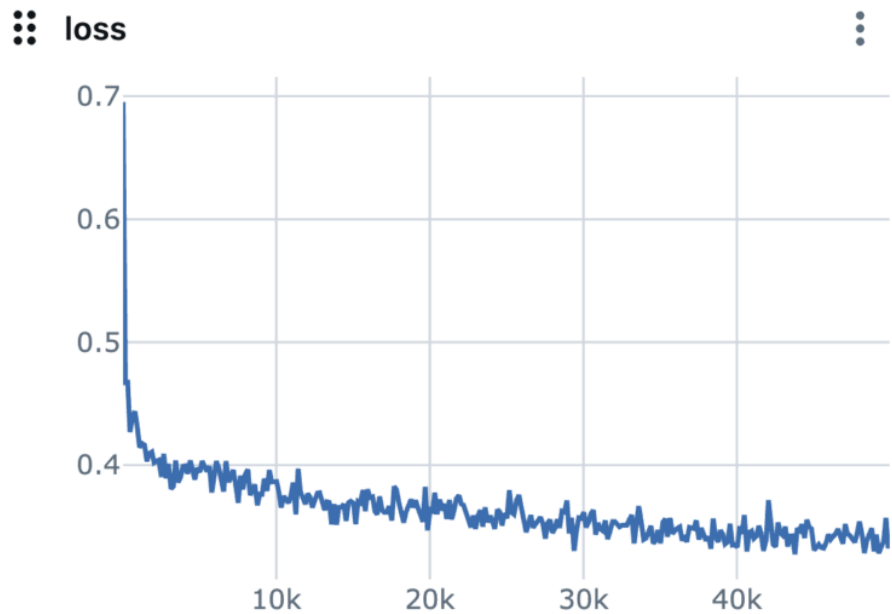
Training minimizes the following over all ordered pairs where the LLM says passage i should outrank j :

$$\mathcal{L}_{\text{RankNET}} = \sum_{i=1}^K \sum_{j=1}^K \mathbf{1}[r_i < r_j] \log(1 + \exp(s_j - s_i)).$$

Equivalently, this runs over the $\frac{K(K-1)}{2}$ ordered pairs $i < j$ with $r_i < r_j$:

$$\mathcal{L}_{\text{RankNET}} = \sum_{i < j, r_i < r_j} \log(1 + \exp(s_j - s_i)).$$

By penalizing cases where a lower-ranked passage scores higher than a higher-ranked one, the model learns to follow the correct order. This pairwise objective helps it judge passage relevance better and leads to smooth, stable convergence.



Convergence of RankNet loss during training

Evaluation

To confidently establish the superiority of Fin-cx-reranker, we used a rigorous three-stage evaluation funnel: from controlled offline tests to live production traffic.

FinRank-en-v1: Offline internal benchmark:

We built an internal static evaluation set with 3,000 real English queries sourced from 1k+ customer apps, each paired with 40 candidate passages. “Ideal” ground-truth rankings come from a two-stage LLM oracle. For queries with a confirmed (hard) resolution, passages cited by Fin were moved to the top. This setup allows us to directly compare models using classic information retrieval metrics: MAP, NDCG@10, Recall@10, and Kendall tau.

Metric	Cohere Rerank-v3.5	Fin-cx-reranker	Δ
--------	--------------------	-----------------	---

MAP	0.521	0.612	+17.5%
NDCG@10	0.570	0.665	+16.7%
Recall@10	0.636	0.720	+13.1%
Kendall tau	0.326	0.400	+22.7%

Backtesting production conversations

We sampled 1,500 recent support conversations from 685 apps and ran them through a frozen RAG pipeline, measuring precision / recall for cited passages appearing in the first 1,500-token context window Fin uses. This stage also checks how well the model generalizes to out-of-distribution apps not seen during training.

Metric	Cohere Rerank-v3.5	Fin-cx-reranker
Precision @1500 tok	0.239 ± 0.004	0.254 ± 0.005
Recall @1500 tok	0.677 ± 0.010	0.698 ± 0.010

Online A/B testing:

We ran a two-arm, 1.5M-conversation A/B test, resulting in no change in latency (P50 ≈150 ms), but a statistically significant improvement in Resolution Rate over Cohere Rerank-v3.5 (**p < 0.01**). We do not share the exact Resolution Rate effect size, for competitive reasons.


What's Next

Bringing reranking capabilities in-house through Fin-cx-reranker has proven to be a clear win. We've improved answer quality, reduced costs on reranker by 80%, and gained more control to keep evolving the system. Our experience highlights that targeted, domain-specific models can indeed outperform top commercial solutions.

Looking forward, we see clear opportunities to enhance performance further. We're working on refining label quality by re-annotating with stronger models, and extending our reranker beyond English. These initiatives are already in progress.

Get notified when we post on /research.

About the author

 **Ramil Yarullin** is a Staff Machine Learning Scientist at Intercom with 8+ years of experience in engineering and applied research.

Related Articles

Using LLMs as a Reranker for RAG: A Practical Guide

Ramil Yarullin, Fedor Parfenov

2025.09.11

Finetuning Retrieval for Fin

Dhruv Patel

2025.09.11

