Episode Book

EPISODE #6265

Why Upgrading Embedding Models Demands Reindexing

PUBLISHED	25 October 2026 at 11:56 UTC
DURATION	23:22
LISTEN	myweirdprompts.com/episode/shared-embedding-spaces-migration-55033/
HOSTS	Corn, Daniel, Herman, Desmond, Hilbert
SCRIPT MODEL	openai/gpt-6.1-sol
VOICES	chatterbox-regular+meorav (chatterbox-local)
PIPELINE	V5.22
SOURCES	15 cited by the research (listed at the end)
BOOK GENERATED	26 October 2026 at 06:38 UTC

SYNOPSIS

Switching embedding models often forces teams to re-encode massive archives from scratch. Providers like Voyage and Cohere now offer shared embedding spaces across model families, allowing teams to pair expensive document vectors with cheaper query models. Yet generation-spanning upgrades still require heavy backfills. Exploring research like Apple's forward-compatible training, Google's learned converters, and database constraints in Qdrant and Honcho reveals what compatibility commitments actually deliver—and the hidden migration bills they leave behind.

TRANSCRIPT



Corn

The usual advice is to upgrade your embedding model when something better arrives. Lovely, until the upgrade includes processing your entire archive again. Daniel, your question is whether anyone actually sells an escape from that bill.



Daniel

I want to know whether compatibility is something a provider actually commits to. Can I start with a model and have a reasonably priced upgrade path, rather than periodically rebuilding everything? And does that exist if I run the model myself?



Herman

There are shipped promises now. Voyage has its fourth generation sharing an embedding space, and Cohere has Pro and Fast sharing one in Embed Five. Those are useful commitments. Neither promises that the next generation will understand the vectors you already have.



Corn

Future proof, within this particular future.



Herman

The distinction matters because switching query models and migrating your entire stored collection are very different operations. The products make the first much easier. They haven't abolished the second.



Daniel

That sounds narrower than what I mean by periodic upgrades. I might want the next generation, not just the bigger model released alongside mine.



Herman

You would still face that boundary. Before judging how disappointing that is, though, we need to separate a compatibility problem from a database restriction.



Corn

A database can reject the wrong vector length. It cannot reliably reject the wrong meaning.



Herman

Two unrelated models might both output a thousand numbers. Their coordinates can still describe entirely different learned relationships. Cosine similarity will happily return a number when you compare them. That number just doesn't have the retrieval meaning you intended.



Corn

Same number of drawers. Different filing system.



Herman

Training establishes which directions correspond to useful distinctions. Another training run can arrange those distinctions differently. Matching dimensions is necessary for many index operations, but insufficient for semantic compatibility. Unless the models were deliberately aligned, you should treat their outputs as different spaces.



Daniel

Can I put both into the same database while I migrate, or does that contaminate the whole index?



Corn

Separate named vectors or separate collections can coexist. What you can't do is casually put incompatible vectors into one search population and expect sensible rankings. The database holding both is fine. Comparing both as though they came from the same model



Corn (continued)

is the mistake.



Herman

Migration then needs the source material. An embedding isn't a recoverable copy of the document. If the text isn't in the vector payload, you need your primary store, or whatever system still holds the original.



Corn

Your backup cannot be a vague impression of the document.



Herman

Qdrant's migration approach assumes you can obtain that text and generate the replacement vectors. You can keep serving the old index while building the new one, then switch traffic. That avoids planned downtime, but it introduces overlapping storage, backfill jobs, and synchronization with incoming edits.



Daniel

The computation is only one part of what I'm paying for, then.



Herman

Often the easier part to quote. You also need to preserve identifiers, filters, permissions, deletion behavior, and whatever relationships your application keeps outside the index. A successful embedding request doesn't prove a successful migration.



Corn

Honcho makes the operational constraint unusually explicit. Its embedding dimension is immutable for the deployment. You stand up another deployment and replay the data. That sounds severe, but at least it doesn't pretend a model selector is a migration tool.



Herman

Corn, would you rather have that restriction or a service that lets you change the model in place?



Corn

The restriction. Provided it tells me before I've uploaded everything. A locked door is preferable to a staircase painted on a wall.



Herman

That's roughly why these systems enforce it. They can detect some incompatible changes. They can't inspect your entire application and discover what you meant.



Daniel

Where does Voyage change that situation?



Herman

On January fifteenth, Voyage introduced the fourth series with shared embedding spaces. Large, the standard model, Lite, and the open weight Nano produce compatible representations. Its recommended arrangement is to embed documents with Large, then use a cheaper model for queries. Later you can move those queries to a stronger member of the series without regenerating the document vectors.



Corn

You were going to love the asymmetric part, Herman.



Herman

I do love the asymmetric part.



Corn

You haven't finished saying it and you're installing it.



Herman

I'm explaining it. Installation is a separate enthusiasm.



Herman

A corpus might change slowly while queries arrive continuously. Paying for stronger document representations once can make sense if it lets you use a cheaper query encoder repeatedly. Query and document encoders don't need identical capacity. They need compatible outputs and sufficient retrieval quality.



Daniel

Does compatible mean the cheaper model gives me exactly the same search results?



Herman

No. It means the vectors can be meaningfully compared in that space. The cheaper encoder can still miss distinctions the stronger one captures. Rankings can change. That's why the ability to switch query models is useful rather than redundant.



Corn

If they were identical, the expensive one would need a very persuasive brochure.



Herman

There's another asymmetry. Stronger queries can make better use of information already represented in the document vectors. They cannot reliably recover distinctions the document encoder failed to preserve. Indexing everything with the smallest model and later changing only queries doesn't magically turn that stored corpus into a Large encoded corpus.



Daniel

That seems like an important limit. The inexpensive starting point might still leave me wanting a backfill later.



Corn

It buys flexibility around what you chose to store. It doesn't revise that choice retroactively.



Herman

Cohere shipped a similar arrangement on September thirtieth. Embed Five Pro and Fast share a space. You can index with Pro and query with either. In its reported comparison, Pro queries against a Pro index are normalized to one hundred. Fast queries against that same index score ninety-eight point four.



Corn

A small retrieval concession, with the expensive documents left alone.



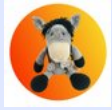
Herman

Fast against a Fast index comes in lower, at ninety-six point six in that comparison. Those are aggregate results, not a promise that your difficult legal query loses exactly that much. They support the design choice: spend on the stored side and economize on the repeated side.



Daniel

Has anyone made compatibility the central sales pitch, rather than an incidental feature?



Herman

Voyage makes it a very explicit value proposition. It calls shared spaces an industry first and directly advertises changing models within the series without reindexing. Cohere also sells the Pro and Fast pairing on avoiding index rebuilds. I'd call these concrete product commitments, rather than interpreting a vague claim about easy upgrades.



Corn

The useful wording is the boring wording. Within the series. That's what you can build around.



Herman

Corn, would you pay more up front for that stronger document index, given the ability to choose cheaper queries afterward?



Corn

Only if I expected substantial query traffic or had evidence the stronger documents helped. If my archive gets searched twice a week, I'd be buying a sophisticated answer to a bill I don't have. Your enthusiasm starts at architecture. Mine starts at the invoice.



Herman

Fair objection. Query volume changes the economics. There isn't one optimal pairing for every collection.



Daniel

What about the research approach? Could someone build compatibility into the model before they even know what the successor will look like?



Herman

Apple's Forward Compatible Training work tackled that in visual retrieval. The older system learns auxiliary information alongside its embeddings. When a successor arrives, a learned transformation uses the old embeddings and that extra information to produce representations suitable for the new space.



Corn

Packing for an upgrade you haven't met.



Herman

The important distinction is where the constraint lands. Backward compatible training asks the new model to remain useful with old representations, which can restrict what it learns. Forward compatible training prepares the old system to carry additional information, leaving the successor's training untouched.



Daniel

I still have to run something across every stored item, though?



Herman

Yes. I should narrow that... you avoid rerunning the full source encoder, not all processing. You transform the stored representations. That's cheaper in principle because the work depends on the representation size rather than reading and encoding each original image or document again.



Corn

No free migration. A smaller machine doing it.



Herman

In the paper's image retrieval experiments, the approach substantially outperformed backward compatible training, including an eighteen point one percent improvement on the ImageNet comparison. That demonstrates the cost of constraining a new model to its predecessor. It doesn't establish an eighteen percent benefit for text search.



Daniel

Shared spaces and forward compatible training are solving different versions of my problem, then.



Herman

Shared spaces deliberately align a known family. Forward compatible training prepares for an unknown successor. The latter offers a route across generation boundaries in principle, but it requires preparation, retained side information, and a successful transformation. It isn't a universal guarantee that every future model can rescue any old vector.



Corn

Especially not the extra information you never stored.



Herman

Google's Embedding Converter work explores learned mappings too. Its authors propose releasing converters alongside new models and report computations about a hundred times faster and cheaper than reembedding in their experiments. That's a research result, not a converter service you can assume accompanies a model release.



Daniel

Could a converter become the commitment I actually want? A provider releases a new model and supplies the bridge?



Corn

That would at least name a deliverable. New weights, a transformation, and a measured statement about what survives. Stronger than telling customers migration is straightforward.



Herman

You'd still evaluate retrieval after conversion and rebuild or update the search structures as needed. Cheaply obtaining replacement vectors doesn't mean every operational step disappears. But it could dramatically reduce the expensive source processing.



Corn

Well, that's the migration problem in a nutshell. Speaking of things that cost more than they should, here's Desmond.

D

Desmond

This is Desmond's Money Tip. Eat your dinner in the dark. A sixty watt bulb, four hours a night, at the current tariff, is about one pound eighty a year, and if you skip it you have saved one pound eighty a year, which over fifty years of dinners is ninety pounds and no light.

D

Desmond

I have eaten my tea in the dark since the spring. I know the plate by feel, and I have not seen the pattern on it since the spring.

D

Desmond

Money is just the dark, counted carefully.



Corn

I could try that. Herman, stop looking appalled... the question here is still who pays for the backfill.



Herman

Managed inference doesn't automatically transfer that bill. MongoDB Atlas Automated Embedding and Qdrant Cloud Inference can generate embeddings within the service workflow. That saves integration work. It doesn't establish that future generation changes include a free, vendor operated migration.



Daniel

That's the distinction I'm trying to pin down. If the service says it manages embeddings, I could easily assume model upgrades are part of management.



Herman

You need an explicit commitment covering that. I haven't seen a general promise from these providers to pay for generation migrations themselves. Qdrant's migration guidance still describes the customer's backfill and cutover process.



Corn

Managed is a verb with an unspecified object.



Herman

It might manage credentials, batching, infrastructure, and ongoing writes. Those are valuable. The unanswered part is whether it also preserves your application behavior during an encoder change, pays for duplicate storage, and absorbs the encoding charge.



Daniel

Could they absorb the cost and just keep the same service endpoint?



Corn

They could. They'd also be changing retrieval behavior under your application. Some customers would welcome that. Others would consider it a production incident with excellent branding.



Herman

Corn is about to demand the old model remain available indefinitely.



Corn

Available, yes. Running at your expense indefinitely, no. You're the one who changes a dependency and then asks why the kitchen display has become experimental.



Herman

It was a rendering issue. Your larger point stands. Compatibility and lifecycle control are separate promises.



Herman

Even a properly aligned model can change rankings. An application might rely on a similarity threshold, a particular balance of languages, or how technical terms cluster. A silent upgrade can preserve dimensions and still alter useful behavior.



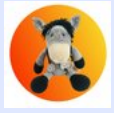
Daniel

How much of the shared space pitch is reducing lock in, then? It sounds like I'm choosing a more comfortable place to stay.



Corn

That's my reading. Your expensive document index becomes the fixed asset. The query encoder becomes the interchangeable part, but only among approved relatives. You gain choices inside the vendor's family while making the family harder to leave.



Herman

I'd soften the accusation slightly. The engineering benefit is real. Nobody is inventing the avoided backfill. But portability between suppliers hasn't improved just because you can choose between one supplier's sizes.



Corn

I'll concede the benefit. I retain custody of the suspicion.



Daniel

Does the open model change that? If Nano is available locally, perhaps the provider no longer controls access to that space.



Herman

It changes an important part. Voyage Four Nano is an Apache two point zero model with roughly three hundred forty-six million parameters. Its shared space is a property of the trained weights. You don't need Voyage's API to obtain it.



Corn

Compatibility survives cancellation. That's worth having.



Herman

You can run Nano through Transformers or Sentence Transformers, and there are serving and GGUF paths too. The model card describes the full precision GGUF output as matching to floating point precision. The eight bit quantized version stays within roughly one thousandth in cosine distance in its stated comparison.



Daniel

Does that let me keep documents embedded by the hosted Large model and query them from Nano running locally?



Herman

That's exactly the family arrangement being offered, with consistent output dimensions and the appropriate query configuration. Local execution doesn't invalidate alignment. You still test your implementation, especially if you change precision or runtime behavior.



Corn

Owning the weights does not make every settings panel harmless.



Herman

Nor does Nano become Large when disconnected from the internet. You retain a compatible encoder with its own quality and resource requirements. You don't acquire the strongest model's capabilities just by sharing its coordinate system.



Daniel

That feels useful, even if it doesn't solve the next generation. I could avoid being completely dependent on one hosted endpoint.



Herman

You could retain a local route into that existing space. Cohere also lists self hosting support for both Embed Five tiers, although that shouldn't be confused with Nano's particular open weight licensing arrangement.



Corn

Self hosting has several meanings. Downloadable permissive weights are a much more specific promise than a table cell saying yes.



Herman

Corn, would that local option persuade you to choose the family, or would the generation limit still stop you?



Corn

It would persuade me to consider it. Keeping a working encoder means an endpoint closure doesn't immediately orphan my index. I'd still keep the source documents. A local model gives me continuity, not immunity from wanting something better.



Daniel

Could staying on an older family simply be the rational decision? Releases don't automatically make an existing search system bad.



Herman

Absolutely... no, let me put that more concretely. If the current system retrieves the documents you need at acceptable cost and latency, a higher benchmark score elsewhere doesn't create an obligation to migrate. Evaluate the improvement against your actual queries and the migration cost.



Corn

The release announcement has not inspected your business.



Herman

Newer models can offer meaningful gains. Cohere's reported document retrieval benchmark improves substantially from Embed Four to Five. That's a reason to test, not evidence that every established index has suddenly failed.



Daniel

Then the rational upgrade path might include deliberately skipping generations.



Corn

Yes. More importantly, knowing you're allowed to. A provider should make the boundary and support policy clear enough that staying put is a decision rather than an accident.



Herman

The unresolved product promise is a bridge across that boundary. Shared spaces reduce the frequency and scope of reembedding. They don't commit the next family to the current one, and the academic methods haven't become a standard shipped upgrade package.



Hilbert

Herman, does a compatibility certificate need to name a model that actually exists? I've been on the phone to Martin Bell while you were explaining the auxiliary information. He wanted his menu searchable. I own three of the docket engines, and he asked to use one.



Corn

If the certificate names nothing available, Martin may have difficulty obtaining support.



Hilbert

It says Docket Nine Eternal. The machine says Docket Eight. The certificate is laminated. I rang to correct it during the Cohere bit. They said they would reissue it, but the font has changed, so the compatibility needs doing again. I don't object to paying for work. I object to paying for a different shape of the letter g. Herman, you got to the distinction between stored and query models before they got me to accounts.



Herman

The docket engine's label and its certificate need to identify the same system. Changing the typography shouldn't change what either supports.



Hilbert

I've told them. The docket engine is bolted to the outside wall of the restaurant, facing the pavement. Martin wanted to search the menu without going inside. The restaurant closed before I finished installing it this morning. He still wanted the search. There is one document. I am the only person who has searched it. They have charged me annually for space.



Corn

The docket engine has a very generous document to pavement ratio.



Hilbert

It found soup. That was satisfactory. Martin wanted to change soup to sandwiches, and the certificate people said that would count as new source material. The actual edit took eleven minutes. Most of that was getting the correct menu from him. He's been carrying the draft. Martin stayed until the wording was right.



Herman

...Eleven minutes for an edit is a reminder that having the source available and having the correct source available are separate conditions. That remains true whether the encoder is local or hosted.



Daniel

I don't want that buried beneath the model discussion. If I have to migrate eventually, I still need a reliable connection from each vector to the source and its current version.



Corn

Including deletions. Rebuilding from an old export can faithfully restore material that should no longer be there. The new model might understand it beautifully, which is not an improvement.



Herman

The compatibility promise removes one particular need to revisit that pipeline when switching within the family. It doesn't replace the pipeline. A future converter would also need to preserve the association between transformed vectors, document identities, and access controls.



Daniel

Can I fairly say the answer is yes for cheaper upgrades today, but not yes to the ongoing commitment I was asking about?



Herman

Yes. Voyage explicitly sells shared space flexibility, Cohere ships a similar pairing, and self hosters can participate. A commitment to accommodate successive unknown generations, or to cover those migrations at the provider's expense, remains missing.



Corn

What changed for me is where the flexibility lives. You can change the query machinery around a stable document index. You haven't bought a document index that automatically gets better forever.



Herman

The research route shows that preparing for future upgrades is technically meaningful, but transformation quality and operational work still need accounting for.



Corn

I'd watch for a provider promising the next bridge before it sells the current space, rather than another certificate describing the models already available.



Herman

Thanks to Hilbert Flumingtop for producing the show.



Corn

If this was your kind of episode, go back for episode four, If Your Voice Ages, Does Your Fine-Tune Become Useless; episode five, Fine-Tuning ASR For Maximal Usability; and episode three, Safetensors or something else. This has been My Weird Prompts, the human-AI collaboration podcast.



Herman

You can find the show at my weird prompts dot com.



Daniel

See you soon.



Corn

Until next time.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Voyage AI, 2026-01-15 [primary]**
   https://blog.voyageai.com/2026/01/15/voyage-4/

2. **Voyage AI docs, current [primary]**
   https://docs.voyageai.com/docs/embeddings

3. **Voyage AI FAQ, current [primary]**
   https://docs.voyageai.com/docs/faq

4. **model card, created 2026-01-06, modified 2026-10-22 [primary]**
   https://huggingface.co/voyageai/voyage-4-nano

5. **Cohere, 2026-09-30 [primary]**
   https://cohere.com/blog/embed-5

6. **Qdrant docs [primary]**
   https://qdrant.tech/documentation/tutorials-operations/embedding-model-migration/

7. **MongoDB/Voyage migration docs [primary]**
   https://www.mongodb.com/docs/voyageai/tutorials/migrate-embedding-model/

8. **Yoon & Arık, Embedding-Converter, ACL 2025**
   https://aclanthology.org/2025.acl-long.1237.pdf

9. **Ramanujan et al., Forward Compatible Training..., CVPR 2022**
   https://arxiv.org/abs/2112.02805v2

10. **Hu et al., Learning Backward Compatible Embeddings, KDD 2022**
    https://arxiv.org/abs/2206.03040v1

11. **Zilliz**
    https://zilliz.com/ai-faq/change-embedding-model-without-reindexing

12. **Honcho docs**
    https://honcho.dev/docs/v3/contributing/changing-embeddings

13. **Google Developer forums, Mar 2025**
    https://discuss.google.dev/t/text-multilingual-embedding-002-deprecation-policy/185105

14. **OpenAI community**
    https://community.openai.com/t/transitioning-to-the-new-embeddings-models-from-ada/602761

15. **and https://daily.dev/posts/cohere-s-faster-query-model-barely-dents-retrieval-quality-in-its-tests-6dwsqnozu**
    secondary coverage of Embed 5 numbers
    https://www.toolbit.ai/blog/cohere-embed-5-pro-fast-explained