Episode Book

EPISODE #6263

Why Numeric Data Lacks a Default Embedding Model

PUBLISHED	25 October 2026 at 11:50 UTC
DURATION	24:00
LISTEN	myweirdprompts.com/episode/numeric-time-series-embeddings-48016/
HOSTS	Daniel, Corn, Herman, Larry, 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

While text and vision have standard embedding models, numeric time-series data lacks a general-purpose retrieval default. Looking across methods reveals the divide between forecasting architectures and similarity search. The discussion covers simple normalized coordinate windows in systems like InfluxDB and Milvus, dimension-reducing transforms such as KDB.AI's temporal search, multimodal financial alignment, and specialized vector indexing in Timescale Vector. When analyzing numbers, calculating exact values and matching shapes require fundamentally different engineering choices.

TRANSCRIPT



Daniel

I want an agent to look through numbers and find patterns, not just calculate totals. Say I give it some stock price history and ask for periods that resemble it. With text, I'd choose an embedding model. What's the equivalent when my entire namespace is numeric?



Corn

Daniel's putting a price chart into the slot where we'd usually put a paragraph. The database can store vectors either way. The awkward part is what turns those numbers into something worth comparing.



Herman

There isn't a canonical, general-purpose numeric retrieval model that fills the role of a popular text encoder. There are useful tools, but they hide several different operations behind the word embedding.



Daniel

I'm interested in that distinction. If something takes numbers and returns a vector, what would make it the wrong tool?



Herman

Its training objective. A forecasting model can be excellent at predicting the next value without putting historically similar cases near each other. Both operations involve internal representations. Only one is explicitly trying to make a useful search space.



Corn

Predicting tomorrow doesn't qualify you to remember Tuesday.



Herman

Especially when similar means similar for a particular decision. Two price windows might share a shape, a volatility regime, or a cause. Those are different retrieval targets, even before we choose software.



Corn

The advice to choose an encoder for your data type quietly assumes there's a menu. Text has one. Images have one. Numeric windows get a kitchen inspection.



Herman

Three things get called numeric embeddings. First, just take a fixed window of observations, normalize it, and use those values as coordinates. No neural network. Second, train an encoder to project a series into a learned space. Third, reduce the window into a compact representation using a transformation.



Daniel

In that first case, the vector really is just the data?



Herman

The transformed data, yes. Suppose you're comparing a day's traffic measurements with earlier days. Each position represents a corresponding observation. Normalization might remove differences in overall level so the comparison emphasizes the daily rise and fall.



Corn

Although removing the level could remove the thing you care about. A spending pattern of ten, twenty, ten resembles a pattern of a thousand, two thousand, a thousand. The bank notices the distinction.



Herman

That's the decision hiding inside preprocessing. Corn, would you want those treated as similar spending patterns?



Corn

For finding recurring behavior, probably. For detecting an alarming week, no. I'd want the pattern and the actual amount available separately. Making the search forget the amount shouldn't make the agent forget it too.



Herman

You can preserve the raw values and metadata alongside the search representation. Retrieval doesn't require throwing away the original records.



Herman

The compact transformation example is KDB.AI's Transformed Temporal Similarity Search. Its method called tsc takes windows of arbitrary length and reduces them to a fixed dimensionality, eight by default. That gives the index a consistent vector size.



Daniel

Does eight dimensions mean eight learned financial concepts?



Herman

No. A coordinate isn't automatically an interpretable concept. Nor does accepting arbitrary window lengths guarantee that every comparison across durations is meaningful.



Corn

Eight dimensions. None labelled rent.



Herman

The learned encoder approach is different again. It needs an objective that tells it which relationships to preserve. That objective, more than the fact that the input contains numbers, determines what similar means.



Daniel

Make the raw-window approach concrete for me. What would the actual pipeline look like without a model?



Herman

There's an InfluxDB and Milvus example using generated traffic sensor readings. The series goes into InfluxDB, gets queried back as windows, normalized, turned into vectors, and inserted into Milvus. A new window retrieves similar historical windows to help classify an anomaly.



Corn

The word vectorize is doing excellent camouflage work there.



Herman

It sounds like inference, but it's arranging observations into a searchable numerical form. The time-series database holds the observations. The vector database holds the representations and references back to them.



Daniel

Would that be a reasonable small-scale starting point, or does it fall apart immediately?



Corn

I'd worry about alignment before scale. If the morning rush starts a little later, corresponding coordinates compare different stages of the same event. You could retrieve a worse behavioral match just because its clock agrees with yours.



Herman

That can happen with pointwise distance. Normalization solves some problems, not shifts in timing. Window duration, sampling rate, missing observations, and the distance measure all matter. A short vector doesn't excuse a long argument about what belongs in it.



Corn

You were visibly preparing that argument, Herman.



Herman

I'm exercising restraint. I haven't said resampling yet.



Daniel

Where does a learned model buy something beyond that?



Herman

A financial retrieval approach builds a shared space derived from CLIP, aligning financial time series with language and visual representations. You can search using a description or a hand-drawn chart. The ambition is to retrieve something like a low-volatility regime without specifying every acceptable movement as a numerical predicate.



Corn

I'd predict you want the sketch interface immediately.



Herman

I do. Wouldn't you?



Corn

Your sketches of a rising market would need ceiling clearance.



Herman

My enthusiasm doesn't alter the axes. The useful part is that people often recognize a pattern they struggle to express as a filter. A learned alignment can translate that recognition into retrieval.



Daniel

That's much closer to what I'm asking. Could I download that kind of encoder and expect it to work on transaction history too?



Herman

Not on that basis alone. The financial chart objective doesn't automatically teach it payment behavior. Transactions include irregular timing, merchant categories, refunds, currencies, and account relationships. Turning them into windows requires choices that a stock chart model may never have encountered.



Corn

A namespace containing only numbers can still contain several incompatible meanings.



Daniel

If there's no standard model to choose, what am I actually supposed to put between the data and the vector database?

L

Larry

Do you lie awake wondering which embedding model fits your numeric data? Then you need Semantrix Numeric Embedding Complex, the only remedy that turns any window of raw numbers into a smooth, queryable latent space in eight dimensions, no menu required. Ask your doctor whether Semantrix is right for your namespace.

L

Larry

Side effects may include mild trend volatility, unexpected dimensionality reduction, loss of SQL filters, spontaneous reranking of personal memories, a persistent sensation that your stock history resembles a hand-drawn sketch, and in rare cases your transactions may embed themselves in unrelated namespaces and refuse to aggregate. Semantrix is not for everyone. If your data type has no menu, consult a vendor.



Herman

The claim that any window becomes a smooth, queryable latent space needs a definition of...



Corn

Herman, leave the medicine alone.



Herman

The practical answer is to choose the similarity operation first. Raw windows are one option. A domain-trained encoder is another. A database's integrated temporal search can handle the transformation and indexing together.



Herman

KDB.AI is unusually explicit about that last option. You pass a raw price window to its temporal search and request the nearest matches. It also supports outlier searches, and multiple query windows in one request.



Daniel

Outliers meaning the least similar windows?



Herman

Its interface exposes that through a negative result count. Asking for negative ten selects outlier search rather than the usual nearest ten. That's an API convention, not negative quantities of results.



Corn

Finally, an honest negative return.



Herman

Its documentation recommends HNSW, IVF, and flat indexes for this use, but specifically warns that the additional product quantization in IVFPQ severely damages result quality. Compression can erase distinctions that matter to the temporal comparison.



Corn

That warning makes the database choice rather less interchangeable. A representation can survive being compressed enough for one task and lose what matters for another.



Herman

It doesn't prove all numeric vectors are uniquely fragile. It does show that this temporal search path needs its own validation, rather than inheriting assumptions from a text-search deployment.



Daniel

Are the other options similarly purpose-built?



Herman

Not all. Timescale Vector combines PostgreSQL and pgvector with a DiskANN-inspired index and time-based filtering. That helps keep vector search near time-organized records. It doesn't hand you a universal numeric encoder.



Corn

Redis has a card-transaction recipe combining search, filtering, similarity, and aggregations. That combination matters more than the presence of a vector column. The transactions remain records you can calculate with.



Herman

InfluxDB plus Milvus separates storage from similarity search. Kinetica has pitched natural-language access to quantitative operational data. Across these offerings, the useful capability often arrives as a database feature or pipeline, rather than a model you download independently.



Daniel

What about the IBM time-series models? Are those the obvious models people would mistake for the answer?



Herman

Granite is the conspicuous family. In the supplied Hub snapshot, the leading TSPulse model had roughly one point seven million monthly downloads. That signals real adoption, but a download count doesn't establish a retrieval interface.



Corn

Popular at the wrong job is still the wrong job.



Herman

The TTM release lists forecasting, adaptation, fine-tuning, multivariate forecasting, external variables, and batch inference. Its internal embeddings support its architecture. They aren't, by themselves, a claim that nearest-neighbor search produces relevant historical cases.



Daniel

Could I extract those internal representations anyway?



Herman

You could experiment. Then you'd have to demonstrate that the distances mean what your application needs. You'd be building a retrieval method, rather than selecting an established one.



Corn

The catalogue searches are also narrower evidence than they first appear. A name search returning nothing for time series embedding doesn't mean nobody has built one. It means you shouldn't mistake an empty shelf under that exact label for a census.



Herman

The stronger point is the landscape taken together. Domain-specific encoders exist. Transformations exist. Forecasting models are popular. None supplies the broadly adopted, general-purpose numeric retrieval default Daniel is asking for.



Daniel

Why hasn't that default emerged? There's an enormous amount of numerical data. It doesn't feel like a small potential market.



Corn

Plenty of the demand is already served by calculation. Ask how much revenue came from Germany last quarter and there may be no stored item containing the answer. Someone has to select the right rows and add them.



Herman

Similarity can't replace that operation. Embedding transaction rows as text doesn't make a nearest-neighbor result an exact total. The industry's usual answer is an agent that generates SQL or operates on a DataFrame, with a calculation engine doing the arithmetic.



Daniel

I'm not expecting a vector to add things up. I want to describe a pattern without having to specify all the filters.



Herman

That demand is real. It just occupies a narrower slice than natural-language access to numeric data as a whole. Some requests need pattern retrieval. Many need computation. Both can sound conversational while requiring different tools.



Corn

SQL loses no sleep over being less conversational.



Herman

Corn, suppose an agent hears spending that resembles this pattern. Would you let it select the normalization itself?



Corn

I'd let it propose one. I wouldn't let that proposal disappear inside a confident answer. Similar monthly shape, similar absolute spending, and similar merchant mix can return different histories. The agent should expose which comparison it made.



Daniel

Could it retrieve candidates broadly and then calculate the exact characteristics afterward?



Herman

Yes. Filters can constrain account, currency, and time. Similarity retrieves candidate windows. Exact calculations describe those candidates. A reranker can then prioritize the relationships you care about.



Corn

Although anything excluded at the first stage stays excluded. Beautiful analysis of the wrong shortlist remains wrong.



Herman

Candidate coverage matters. You can use multiple retrieval routes rather than trusting a single geometry to capture everything.



Daniel

There's still a model-training question underneath this. Why is it harder to learn numeric meaning than language meaning?



Herman

Text provides enormous quantities of context, paired passages, paraphrases, and descriptions. Numeric series need some equivalent account of relevance. A normal network-traffic pattern doesn't look like a normal heart-rate pattern. Even within finance, similar shape might mean different events.



Corn

A sudden flat line is excellent for one chart and urgent for another.



Herman

My pediatrician instincts refuse to let that pass without checking the sensor. That's part of the problem, actually. The meaning depends on the measurement process, not merely the sequence.



Daniel

Could very large training sets teach those distinctions, with enough domain information attached?



Herman

They could help. The thing is... once you attach domain information, units, context, and outcome labels, you're specifying the task more richly. You're no longer relying on numbers alone to reveal a universal meaning.



Corn

The numbers don't tell you which differences you're willing to ignore. That comes from what you want to do with the match.



Herman

There's a smaller version of this problem when people combine text vectors with numerical features. Just concatenate them, and whichever block has the larger magnitude can dominate the distance.



Daniel

Even if the numerical features are the important part of the query?



Herman

Importance doesn't emerge automatically from concatenation. One worked example has a text block with roughly ten times the magnitude of the numeric block. That is an example of a scaling problem, not a universal constant of embeddings.



Corn

The merchant description gets a megaphone. The amount gets a chair.



Herman

Normalizing each block and weighting text, categories, numeric features, and domain features separately can help. The weights then need tuning against actual relevance. A sensible-looking allocation is a hypothesis until you test it.



Daniel

That sounds like a lot of manual work hiding behind choose an embedding model.



Corn

The model selection advice skips the part where someone defines the task. Text systems skip it too, but numeric data makes the omission harder to overlook.



Herman

You defend the old, inspectable option before I've even finished explaining the learned one, Corn. Would a retrieval benchmark ever persuade you to favor the encoder?



Corn

Show me better matches for the actual use. I object to graduating the model because it owns a latent space.



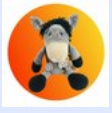
Herman

Then you'd appreciate READ-Bench. It compares historical-case retrieval across diagnostic time-series datasets, with relevance based on shared fault type rather than merely similar signal shape. Under its common channel-independent interface, pretrained representations showed no statistically detectable advantage over strong classical and symbolic baselines for search alone.



Daniel

Hm. Was the semantic target deliberately different from what the curves looked like?



Herman

Yes. That distinction is what makes it informative. A useful historical case may be one with the same diagnosed fault, even if its observations aren't the closest visual match.



Corn

The shape can be a clue without being the answer.



Herman

What improved results substantially was a little resolved-case supervision at reranking. A Gaussian-process reranker used neighboring labeled cases to adjust which candidates rose to the top. In that setup, this helped more than more sophisticated representations or language-model reasoning.



Daniel

So some feedback saying these were useful matches could matter more than replacing the embedding model?



Herman

In that benchmark, yes. I wouldn't generalize it into a law for every financial search. It does challenge the assumption that representation sophistication is always the most productive investment.



Corn

You've been saving another classical method, haven't you?



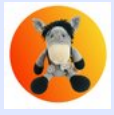
Herman

Dynamic time warping.



Corn

There it is.



Herman

Their winning system combines a learned retrieval leg, scored using deviations from normal behavior, with a dynamic time warping leg. It merges the rankings rather than requiring their raw distance scores to be directly comparable.



Daniel

What does dynamic time warping catch that the raw-window vector might miss?



Herman

It aligns sequences while allowing local stretching and compression along time. Two rises and falls can match even if one unfolds more slowly. It has constraints and computational costs, but it directly addresses the timing issue Corn raised earlier.



Corn

Similarity with a flexible clock. Useful until the clock becomes so flexible that unrelated events agree.



Herman

Alignment rules still need discipline. No method removes the need to define acceptable similarity.



Herman

Another research direction, TRACE, grounds series representations in aligned text. Descriptions provide meaning that the numerical shape alone doesn't specify. That's promising, but notice what supplies the semantic supervision.



Daniel

Language comes back in through the training signal.



Corn

Even when the namespace stays numeric.



Hilbert

Herman, does a price window have to belong to the person who pays for the search? I'm dealing with Martin Bell this month. He wants spending histories that resemble his own, but he won't provide his transactions. He hires me to live his proposed bank statement for thirty-one days. I'm currently buying his Wednesday groceries.



Herman

You can search using a constructed example. It needs to be labelled as constructed, though. Otherwise the results make the example look more authoritative than it is.



Hilbert

That's on the invoice. Synthetic household, actual purchases. There are forty transactions. Forty-two, counting refunds as separate events. I count a returned item if it crosses the counter. Martin wants the daily amounts normalized. He also wants me to preserve the fact that he considers lunch over twelve pounds a personal defeat. Those instructions remain in the same email.



Corn

His lunch threshold survives normalization by correspondence.



Hilbert

He has the spreadsheet on his laptop. He sends me screenshots because he says exporting it changes the figures. The screenshots arrive sideways, which is awkward with an anteater's claws on the rotation button. Corn, your distinction between shape and amount takes less space than his instructions. I'm putting it on the next invoice.



Herman

A screenshot also makes the actual values harder to recover. You need the observations if you're comparing the windows reliably.



Hilbert

My cousin Neil, who builds retrieval systems and isn't to be trusted, says the expensive part is getting Martin to choose what counts as a match. He says a model can return candidates all afternoon while Martin changes his mind about whether a refund belongs to the original week. Neil wants payment for a relevance review. Martin agrees to pay for exactly twenty examples. He spends the first appointment moving a refund from Thursday to Monday.



Daniel

That relevance review sounds like the bit I couldn't get simply by downloading a model. Could an agent ask me those questions as part of setting up the search?



Herman

It could show contrasting matches and ask which distinction matters. Same shape at different amounts, or same amount with different timing. Those answers could guide normalization, retrieval routes, or reranking. That's a plausible interaction design, not a capability guaranteed by the database.



Corn

It would also make ambiguity visible before the agent produces a polished explanation of a match you never wanted.



Daniel

I'm getting a clearer separation now. There's the engine that finds candidates, then the question of whether those candidates mean anything for my request.



Herman

Both deserve evaluation. A fast index tells you how efficiently you can retrieve neighbors. It doesn't establish that neighboring windows represent the cases you meant.



Corn

The part worth keeping is that numeric similarity is a chosen relationship. Raw windows, learned encoders, and compact transformations preserve different things. None gets to define relevant by default.



Herman

Better labels and reranking can improve that relationship without replacing the representation at all.



Corn

Will a standard numeric encoder emerge once enough of those relevance decisions are captured, or will the value remain with databases that keep the windows, filters, and calculations together?



Herman

Thanks to Hilbert Flumingtop for producing the show.



Corn

If you enjoyed this one, try episode nine, Benchmarking Custom ASR Tools - Beyond The WER; episode eleven, How Does Fine Tuning Work Anyway; and episode five, Fine-Tuning ASR For Maximal Usability. This has been My Weird Prompts, the human-AI collaboration podcast.



Herman

If you enjoyed it, leave us a review wherever you're listening.



Daniel

See you soon.



Corn

Until tomorrow.

SOURCES

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

1. **KDB.AI docs, Transformed Temporal Similarity Search [primary]**
   https://code.kx.com/kdbai/latest/use/transformed-tss.html

2. **KDB.AI homepage, TSS search examples [primary]**
   https://kdb.ai/

3. **IBM model card, created 2026-05-21, modified 2026-07-02 [primary]**
   https://huggingface.co/ibm-granite/granite-timeseries-ttm-r3

4. **Anyscale RAG with structured data docs [primary]**
   https://docs.anyscale.com/rag/structured-data

5. ***Multi-Modal Financial Time-Series Retrieval Through Latent Space Projections* (full text read)**
   https://arxiv.org/abs/2309.16741

6. **READ-Bench, submitted 2026-09-26, rev. 2026-10-08**
   https://arxiv.org/abs/2609.32123v2

7. **TRACE, submitted 2025-06-10, rev. 2026-01-31**
   https://arxiv.org/abs/2506.09114v3

8. **MemoriesDB, submitted 2025-11-09**
   https://arxiv.org/abs/2511.06179v1

9. ***Embeddings at Scale*, ch. 10**
   https://snowch.github.io/embeddings-at-scale-book/chapters/ch10_advanced_embedding_patterns.html

10. **InfluxData, InfluxDB + Milvus tutorial**
    https://www.influxdata.com/blog/time-series-influxdb-vector-database/

11. **SiliconANGLE, 2024-03-18**
    https://siliconangle.com/2024/03/18/kinetica-ramps-rag-generative-ai-workloads-empowering-real-time-operational-data/

12. **LlamaIndex Timescale Vector docs**
    https://developers.llamaindex.ai/python/framework/integrations/vector_stores/timescalevector/

13. **DEV.to, TimescaleDB CAGGs + pgvector**
    https://dev.to/satanic47/high-throughput-financial-timeseries-with-timescaledb-caggs-pgvector-cosine-search-30fa

14. **Redis card-transaction search recipe**
    https://github.com/redis-developer/redis-ai-resources/blob/main/python-recipes/feature-store/01_card_transaction_search.ipynb

15. **RAG Over Structured Data: Text-to-SQL vs Vectors**
https://ai-tldr.dev/learn/rag/advanced-rag/rag-over-sql-databases/