Episode Book

EPISODE #6304

Why Semantic Desktop Search Struggles to See Files

PUBLISHED	27 October 2026 at 18:35 UTC
DURATION	22:17
LISTEN	myweirdprompts.com/episode/semantic-desktop-file-search-72169/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	openai/gpt-6.1-sol
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.22
SOURCES	21 cited by the research (listed at the end)
BOOK GENERATED	28 October 2026 at 09:53 UTC

SYNOPSIS

Semantic vector search is finally arriving on operating systems, but finding files locally remains surprisingly tricky. Operating system updates to Apple Spotlight and Windows Search now bring local embedding models to personal computers, yet application boundaries, missing OCR, hardware gates, and cloud placeholders create blind spots. Examining native implementations alongside independent tools like Lens, PixaFind, and Omni reveals why local indexing is expensive, invisible, and deeply fragmented across different platforms.

TRANSCRIPT



Corn

If my computer can describe a photograph, why can it still lose a document sitting on its own drive? Daniel takes our run on search onto the desktop today. Vector search is everywhere in AI tools, but finding your own files still often means remembering what you called them. He wants to know whether genuine semantic desktop search has arrived, what you can actually use, and why getting here took so long.



Herman

It has arrived, inside Spotlight and Windows Search, although owning the search box doesn't necessarily mean owning the feature. Hardware, application support, and whether your computer has finished building its index all matter.



Corn

My file now has eligibility requirements.



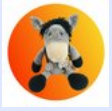
Herman

The distinction matters for Daniel particularly because he uses Linux, where there are independent projects but no single operating system rollout to point at. Across platforms, you can buy or install real vector search today. You cannot assume it sees everything your ordinary file browser sees.



Corn

That's the promise I want to examine. I don't need my computer to understand the entire universe. I need it to remember the PDF I downloaded on purpose.



Herman

A smaller universe. Surprisingly difficult administration.



Corn

We can keep the embedding explanation short. Exact search knows the letters in cow. Semantic search can associate cow with cattle, bovid, ungulate, heifer. Howard Oakley describes that experience as more like a thesaurus. The useful step beyond synonyms is finding something whose wording I never guessed.



Herman

The model produces numerical vectors for your query and the material you're searching. Related content should land near one another in that multidimensional space. An image can participate too, if the model supports it. You could describe a red bicycle beside water without the photograph having those words attached.



Corn

Whereas fuzzy matching might rescue me from misspelling bicycle, but won't necessarily get me from bicycle to a photograph called Thursday.



Herman

Right... spelling tolerance and meaning are different operations. Traditional desktop indexes extract filenames, metadata, and text tokens. An inverted index connects those tokens to documents. There can be prefix matching, case handling, and other rules, but the words still do the work.



Corn

I want both. If I remember an invoice number, I'd rather the computer not explore its emotional associations.



Herman

You should get both. Semantic indexing can sit alongside lexical indexing, and Windows explicitly does that. Meaning helps when memory is approximate. Exact matching helps when the identifier matters. A search result about a similar invoice is quite a different outcome from finding your invoice.



Corn

Especially to the person expecting payment.



Herman

Also, content is often split into chunks before embedding. A paragraph about cancellation terms can be more useful than one vector trying to represent a whole contract. Images, scans, and video introduce different extraction work. Before any clever ranking happens, something has to read the file.



Corn

That's where the universal search pitch starts quietly dropping nouns. Everything becomes everything we can open, decode, and index.



Herman

Would you accept narrower coverage if the search box showed you precisely what was included?



Corn

Yes. Tell me it searches these folders and these applications. I can work with a boundary. An empty result pretending to speak for my entire computer is much harder to work with.



Herman

Apple makes that boundary especially important. At WWDC this year it announced rebuilt search infrastructure behind Spotlight, Photos, and Mail for macOS twenty-seven, Golden Gate. Then on September fourteenth it began rolling out Siri AI in beta, integrated into Spotlight on the Mac.



Corn

The search box I already knew, now doing something I might not know it does.



Herman

Apple's example is asking what a family member suggested for an upcoming visit. Siri finds a recipe mention in messages, retrieves cooking instructions from email, then adds ingredients to Reminders. Retrieval across applications becomes part of an action, rather than a separate expedition through folders.



Corn

I'd settle for locating dinner before it starts purchasing dinner.



Herman

That example depends on the participating applications exposing usable information. The semantic path uses per-app Core Spotlight indexes, stored locally in the user's metadata area. That's different from simply turning the old volume-wide Spotlight index into one enormous vector database.



Corn

An application has to volunteer its contents?



Herman

It has to implement the support. The operating system cannot reliably infer every application's internal records and permissions. Oakley's phrase is that it can't happen by magic. Per-app indexing gives applications a way to supply meaningful items, but an application that doesn't participate leaves a coverage hole.



Corn

A remarkably technical way to lose a note.



Herman

It also helps preserve application boundaries and user privacy. Searching records supplied by an application is different from treating every private database on the drive as loose public text. The cost is that the whole desktop doesn't become semantically



Herman (continued)

searchable just because the operating system gained a model.



Corn

For the cow example, then, I'd expect related terms, not just an exact word match. More like using a thesaurus to get into the right neighbourhood.

L

Larry

Are you tired of your desktop search returning nothing but filenames and keywords, like a thesaurus with the pages glued shut?

L

Larry

Introducing Semantic Slorp, the only desktop search tool that embeds every file, every photo and every forgotten grocery list into a nine thousand dimensional concept space, so when you type cow it proudly returns cattle, bovid, ungulate, heifer and a blurry picture of your neighbor's goat. But wait, there's more. Semantic Slorp works with zero apps, because it cannot actually read any of them, and it runs entirely on device, meaning it runs entirely on your device, right now, at forty percent of the speed of a normal search. Order in the next eleven minutes and we'll throw in a second empty index absolutely free. Results not guaranteed, goats not included, cow to bovid conversion may vary, do not use Semantic Slorp while operating a thesaurus, side effects include optimism, disclaimers read at speed, see nothing for details.



Herman

Nine thousand dimensions wouldn't establish whether...



Corn

Herman, leave the goat. Microsoft's coverage boundary is different.



Herman

Windows semantic indexing runs locally alongside traditional indexing on Copilot+ PCs. The rollout began with Windows Insiders last August, initially on Snapdragon machines, with AMD and Intel following. By February it was documented as live. The hardware gate is the Copilot+ class of machine with an NPU.



Corn

You are about to defend buying a new computer to find the contents of the old computer.



Herman

I was going to defend efficient inference hardware.



Corn

You always arrive at the shop by the scenic route.



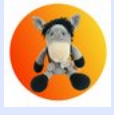
Herman

Fair. Existing computers can run embedding models too. The hardware requirement describes Microsoft's supported product, not a law of physics. An NPU provides a power-efficient place to put repeated local AI work.



Corn

That's a useful distinction. People hear on-device AI and assume their perfectly capable processor has suddenly become illiterate.



Herman

Supported documents include PDFs and Office files, and there are common image formats too, across six languages. I wouldn't turn those lists into a promise about every file extension. More importantly, the documented semantic scope is local files in indexed locations. Cloud storage support, including OneDrive, was described as coming in a future flight.



Corn

A cloud placeholder looks like a file until someone needs its contents.



Herman

Yes... the file browser's view and the indexer's available material can differ. Recall is adjacent but separate. That's the opt-in screenshot history you search by describing something you remember seeing. Ordinary semantic file indexing is enabled by default on eligible PCs.



Corn

Separate consent, separate corpus. I'd hate to diagnose my document search by accidentally discussing screenshots.



Herman

The delay wasn't just waiting for a sufficiently clever model. Parsing years of files, generating embeddings, updating them, respecting deletions, and doing it without draining the battery is substantial work.



Corn

Web search has that bill too. It just doesn't put the bill in my laptop bag.



Herman

Precisely... no, the better distinction is who schedules and pays for the computation. A service operates its indexing infrastructure. Your desktop has to build your private index while remaining your desktop. On macOS the work is throttled toward idle, cool, on-power conditions.



Corn

Finally, software with a sensible attitude toward activity.



Herman

Corn, I knew you'd endorse a machine that waits for everyone to stop touching it. Input marks the Mac active, and it can wait about three minutes after you stop. Media playback can delay the work for hours.



Corn

I endorse the waiting. I object to concealing whether anything happened.



Herman

Large photo libraries and years of mail can take three or four nights plugged in. That means early impressions may reflect an unfinished index rather than the model's retrieval quality. There's no clear progress bar to settle the matter.



Corn

Four nights, no progress bar. Even I provide breathing.



Herman

That's why an immediate search after installation tells you less than you'd expect. The front end may be ready long before the collection behind it is.



Corn

Independent tools can at least let you choose a folder instead of waiting for every application vendor. That's where I'd look on Linux. Although I suspect your answer is going to involve installing several things that all describe themselves as the only thing I need.



Herman

You know my download folder too well. For Daniel, PixaFind and Lens for GNOME are concrete starting points. PixaFind offers offline search on Apple silicon Macs and Linux. Lens combines local embeddings, OCR, and mail synchronisation, with local models through llama.cpp.



Corn

OCR matters more than another adjective on the search button. A scanned document can contain exactly the words I'm typing and still be invisible if nobody extracted them.



Herman

Rememex is another project, advertising support for more than a hundred and twenty file types, OCR, and an MCP server. DigUp has a Mac version with Windows and Linux betas. IntelliFile combines a vector index with SQLite's text search. These are real implementations, but their support, packaging, and coverage vary.



Corn

I wouldn't give Daniel a shopping list and pretend every item is interchangeable. Lens is worth examining for a GNOME desktop and mail. PixaFind is worth examining for offline visual search. Then the question is whether the specific files he cares about are actually supported.



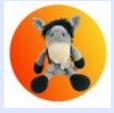
Herman

Would you choose one of those over the built-in feature on a supported Mac?



Corn

Only for something the built-in feature doesn't cover, or because I need visibility into the index. I don't want a second search system merely to have a second search system. You already provide that household service.



Herman

I provide comparative testing.



Corn

You provide duplicate menu-bar icons.



Herman

Omni is the Mac project I'd particularly distinguish. It searches text, images, audio, and video in a shared vector space, on-device on Apple silicon. It runs its embedding model through MLX-Swift and exposes both an HTTP interface and an MCP endpoint.



Corn

Meaning another tool can search the collection, rather than a person having to use its window.



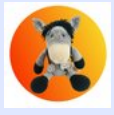
Herman

Yes. Its initial search model download is about one point eight gigabytes. The application code is Apache licensed, while the model weights have a separate noncommercial licence. That distinction matters if someone wants to build a commercial service around it.



Corn

Open source has small print too. It just uses different fonts.



Herman

It's also a small project, with roughly a couple of hundred GitHub stars, not evidence of broad deployment. There is a workshop paper associated with it, but neither a paper nor a star count tells us whether it will find your missing contract.



Corn

What about something I can simply buy?



Herman

Synology Deep Search offers local AI search across folders and drives for forty-nine dollars and ninety-nine cents, once, with a thirty-day trial. Desktop Docs is a Mac option focused on photos and videos using CLIP embeddings. Its move from Electron to Rust and Tauri reduced the app from about a gigabyte to a hundred and seventy-two



Herman (continued)

megabytes.



Corn

They found most of the application and deleted it.



Herman

The claimed indexing improvement is more consequential. A thirty-eight-minute video reportedly indexed in around three minutes, rather than over ten. That's the kind of engineering that makes maintaining a personal media index tolerable.



Corn

Provided the video search finds the moment I want. A fast index of the wrong visual details is still a very punctual disappointment.



Herman

Poly aims at a broader file browser experience, calling itself Cursor for Files. Its model supports retrieval down to a paragraph, a page, or a moment in media. There is a hundred-gigabyte free tier and a two-terabyte tier at ten dollars a month.



Corn

That sounds closer to the independent general-purpose product Daniel is imagining.



Herman

The catch is local desktop file support was still on its roadmap at launch. It belongs in the landscape, but I wouldn't present that launch as proof that it could already search your existing local drive.



Corn

Desktop search, desktop pending.



Herman

I don't see a clear major independent vendor offering the universally supported, cross-platform version people imagine. There are operating system features and a fragmented collection of smaller tools. Some of those tools may be excellent for a particular collection.



Corn

There is also an awkward business problem. I lose one important PDF occasionally. Asking me for a subscription every month means competing with the interval in which I haven't lost it.



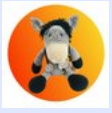
Herman

A specialist can still earn its price through media search, connectors, automation, or better indexing controls. Basic natural-language retrieval alone gets harder to sell when it is already in the operating system.



Corn

I'd pay more readily for a trustworthy coverage report. Which folders were read? Which files failed? Is this result based on the contents or just the filename? Those answers would change how I use the thing.



Herman

How would you decide it had earned your trust? Suppose it finds something impressive on the first try.



Corn

I'd take a few files I know exist, describe them without using their actual wording, and see what happens. Include a boring scan, a long document, and something recently moved. Not a scientific benchmark, but at least the tool doesn't get to choose its own demonstration.



Herman

That's the recall problem. A result can be relevant and still leave several equally important results undiscovered. Precision tells you about the quality of what came back. Recall asks how much of the relevant material came back.



Corn

The triumphant demo shows the wedding photograph. It doesn't show the other wedding.



Herman

Oakley flags exactly that concern: unexpected hits will impress us, while assessing what semantic search misses will be harder. There isn't an established desktop false-negative measurement we can point to and use across these products.



Corn

An empty result could mean wrong words, unsupported format, unfinished indexing, an application that didn't participate, or simply no such file. Those require different responses from me.



Herman

Better infrastructure could fix some of that. Adding a semantic layer alone cannot repair an item that never entered the index. That's the unresolved concern behind Spotlight's rebuild, given its history of occasionally failing even straightforward keyword searches.



Corn

I'd rather see fewer glowing answers and one plain sentence saying that a folder wasn't searched.



Herman

Local processing also deserves a narrower privacy claim. It avoids sending the collection to a remote service, but the local index remains sensitive. A compromised computer may expose information through its index as well as through its original files.



Corn

Especially when the new feature makes forgotten material easy to discover.



Herman

Enterprise administrators need to understand the defaults, disablement controls, and how indexing interacts with data-loss prevention. Apple's application boundaries constrain coverage. Microsoft's default-on approach makes discoverability easier but gives administrators another store of sensitive information to account for.



Hilbert

Corn, would you search for a recipe by the dish, or by the man who keeps asking you for it? Basil wanted the handwritten dumpling instructions from a folder of scans. While you were discussing indexing, I tried both. You've made me suspicious of any result that arrives too confidently.



Corn

I'd use the dish first. Basil may have asked for several dinners.



Hilbert

He asked for this one. I searched for that thing with the thing. The system returned a scan called Parking Objection Final. It was the recipe. I renamed it during a dispute over flour deliveries. I didn't remember scanning it. The paid search app had failed to find it last



Hilbert (continued)

month. That cost me twelve pounds, which they still have.



Herman

That phrase supplies very little meaning. Something else in the context, or the ranking of a small collection, must have helped. I wouldn't expect the same query to work reliably.



Hilbert

It was near the top. I opened it. Basil was waiting on the telephone, so I read the instructions. He runs a dumpling stall inside a document scanner. He says the lid keeps the steam in. I told him the recipe says to leave room between them, which he cannot do during copying hours.



Corn

His lunch rush conflicts with office services.



Hilbert

He wanted the page sent over. The ordinary search found it in seconds once I typed Parking Objection. Before that I had spent sixteen minutes finding the scan that renamed files by content, because I didn't trust another automatic name. Herman, you explained chunking more quickly than its help page explained the rename button. Basil has the correct recipe now. He needs it for tomorrow's batch.



Herman

...Renaming is another point where retrieval and file management meet. A semantic result can recover an item without making its filename any more useful the next time.



Corn

Keep the original name visible, let me add a description, and don't quietly rewrite either. Finding it once shouldn't make the next search harder.



Herman

There's a related detail we haven't touched: Omni offers an optional OCR model download larger than its search model. That's a reminder that reading the collection can demand more machinery than searching the vectors afterward.



Corn

The search box gets the applause. The scanner does the paperwork.



Herman

That extraction work is also where a specialised tool can still distinguish itself from an operating system feature.



Corn

I'd watch whether independent tools make the second search easier, not merely the first discovery spectacular. The opening is coverage you can inspect and failures you can understand. A default search box can be free and still leave room for that.



Herman

Our thanks to Hilbert Flumingtop for producing the show.



Corn

If this was your kind of episode, go back for episode two, Local STT For AMD GPU Owners; episode ten, How ASR Went From Frustration To … Whisper Magic; and episode twelve, The AI Breakthrough. This has been My Weird Prompts, the human-AI collaboration podcast.



Herman

If you enjoy the show, leave us a review wherever you listen. It helps other curious people find us.



Corn

See you tomorrow.

SOURCES

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

1. **Apple Newsroom, updated 2026-09-14. [primary]**
   https://www.apple.com/newsroom/2026/09/siri-ai-a-profoundly-more-capable-and-personal-assistant-is-here/

2. **Apple Support, What's new in macOS 27. [primary]**
   https://support.apple.com/guide/mac-help/whats-new-in-macos-27-apd07d671600/mac

3. **Windows Insider Blog, 2025-08-20. [primary]**
   https://blogs.windows.com/windows-insider/2025/08/20/copilot-on-windows-semantic-search-and-new-homepage-begin-rolling-out-to-windows-insiders/

4. **Microsoft Support, Recall. [primary]**
   https://support.microsoft.com/en-us/windows/ai/ai-features/retrace-your-steps-with-recall

5. **Microsoft Windows Learning Center. [primary]**
   https://www.microsoft.com/en-us/windows/learning-center/find-files-fast-with-improved-search

6. **Omni, Apache 2.0, created 2026-06-01; paper arXiv 2608.05543 (NeurIPS 2026 Workshop on On-Device Intelligence). [primary]**
   https://github.com/hanxiao/omni-macos/

7. **Synology Deep Search product page. [primary]**
   https://www.synology.com/en-us/products/deep-search

8. **PixaFind. [primary]**
   https://pixafind.com/

9. **Rememex, created 2026-02-15. [primary]**
   https://github.com/illegal-instruction-co/rememex

10. **Lens for GNOME. [primary]**
    https://github.com/cwittenberg/lens-for-gnome

11. **IntelliFile v2.4.0. [primary]**
    https://intellifile.vercel.app/

12. **NeuCockpit v1.0. [primary]**
    https://neuron.zero-x.live/

13. **DeskSearch, created 2026-03-22. [primary]**
    https://github.com/wshuai190/desksearch/

14. **The Eclectic Light Company (Howard Oakley), 2026-06-28.**
    https://eclecticlight.co/2026/06/28/last-week-on-my-mac-spotlight-on-semantics/

15. **Let's Data Science, 2026-07-02 (citing MacRumors/WWDC sessions).**
https://letsdatascience.com/news/apple-rebuilds-search-infrastructure-powering-spotlight-and-b37590b7

16. **macworld.info, undated, on macOS 27 Golden Gate indexing.**
https://macworld.info/article/spotlight-indexing-macos-27-slow/

17. **9to5Mac, 2026-06-08.**
https://9to5mac.com/2026/06/08/new-siri-whats-new/

18. **WindowsForum, 2026-02-24, summarizing Microsoft docs.**
https://windowsforum.com/windows-news.4/windows-introduces-on-device-semantic-indexing-for-copilot-pc-search.403164/

19. **Launch HN: Poly (YC S22), 2025-11-20.**
https://news.ycombinator.com/item?id=45995394

20. **Show HN: Desktop Docs rewritten in Rust, 2025-05-28.**
https://news.ycombinator.com/item?id=44118023

21. **DEV Community, 2026-10-21.**
https://dev.to/dd8888/digup-searches-local-media-documents-and-code-on-device-4n68