Episode Book

EPISODE #6710

Why BM25 Still Beats Neural Search on Keywords

PUBLISHED	20 November 2026 at 14:32 UTC
DURATION	28:29
LISTEN	myweirdprompts.com/episode/why-bm25-ranking-works-18376/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	google/gemini-4-carbon
VOICES	omnivoice
PIPELINE	V5.36
BOOK GENERATED	20 November 2026 at 21:20 UTC

SYNOPSIS

Counting matching words is easy, but ranking them accurately without understanding meaning remains surprisingly resilient. Behind the thirty-year reign of BM25 lies a hand-fitted saturation curve designed to prevent single-word repetition from hijacking queries, paired with inverse document frequency from information theory. Examining document length normalization reveals how search engines mathematically distinguish thorough discussions from wordy tangents across thousands of conversational transcripts.

TRANSCRIPT



Corn

Why is a formula from the nineteen nineties, with zero understanding of what any word actually means, still the baseline every billion-parameter embedding model has to beat?



Herman

And losing to it, half the time, depending on what you type in.



Corn

Daniel sent in the follow-up to our inverted index episode. Last time we built the lookup table over the My Weird Prompts transcript archive, about six thousand seven hundred episodes, so you type a word and get back the exact set of episodes where it appears. Now Daniel wants the next primitive in the series: once you have that pile of matching episodes, how do you put them in order using nothing but keyword counts? No semantics, no vectors, just term frequency, inverse document frequency, length normalisation, and BM25. Why has that formula survived thirty years, and where does it fall apart on a conversational transcript archive like ours?



Herman

I love that we are forcing ourselves to stay strictly blind to meaning today. When you build an inverted index, what comes back is an arbitrarily ordered set, usually sorted by document ID—which for us just means episode number. You type in "lithium," you get four hundred episodes in chronological order, and episode twelve where I mentioned a watch battery in passing sits right above a forty-minute deep dive from last month.



Corn

Chronological order is great if you remember Tuesday and terrible if you remember anything else.



Herman

Ranking is the step that turns an index into an actual search engine. What surprises people who came into tech during the transformer wave is that the reigning champion of count-based ranking isn't some dusty museum piece. BM25, Best Matching twenty-five, comes out of the probabilistic retrieval framework Stephen Robertson and Karen Spärck



Herman (continued)

Jones developed in the seventies and eighties. They implemented it in the Okapi system at City University in London, tuned it at the TREC competitions in the mid-nineties, and Robertson and Hugo Zaragoza didn't publish the definitive monograph laying out the thirty-year arc until two thousand and nine. Yet in retrieval papers coming out this month, November two thousand twenty-six, BM25 is still the benchmark on table one.



Corn

Wait, why twenty-five? Were there twenty-four worse ones sitting in a drawer in London?



Herman

Unironically, yes. Not twenty-four sequential failures, they had BM1, BM11, BM15... numbering experimental weighting schemes inside Okapi as they tested ways to combine term counts and document lengths. BM11 and BM15 turned out to be two extreme opposite assumptions about document length, and when they blended them into one tunable formula for TREC-3 in nineteen ninety-four, they called it BM25.



Corn

Naming your masterpiece after the twenty-fifth spreadsheet tab is deeply honest engineering.



Herman

And the reason we have to pull BM25 apart before we touch meaning or synonyms later in the series is that it's a pure bag-of-words model. It doesn't know word order, or whether two query words appeared in the same sentence or forty minutes apart. All it has are three numbers you can count with a tally stick: how many times the word appeared in this episode, how many episodes across the archive contain the word at all, and how long this episode is compared to average.



Corn

Let's take the tally stick to the first one. Term frequency. If I search our transcripts for "archery," and episode four hundred has it once while episode nine hundred has it thirty-seven times because nobody stopped you in time, episode nine hundred wins. Why isn't that the whole algorithm?



Herman

Hey, when draw-weight physics matter, they matter. Think about what happens if the score is completely linear with term frequency. Suppose a user types "Mongolian archery." Episode A mentions "Mongolian" ten times and "archery" ten times. Episode B never mentions "Mongolian" once and mentions "archery" twenty-five times. Under pure linear term frequency, Episode B gets twenty-five points and Episode A gets twenty. The episode that ignored half your query beats the one that covered both halves thoroughly, just because I said "archery" five extra times.



Corn

Because twenty-five is bigger than twenty, and a linear sum can't tell the difference between covering two ideas and obsessing over one.



Herman

And Robertson and Zaragoza have a clean theoretical name for why linear counting is wrong: "eliteness." Think of eliteness as a hidden binary switch for every word-document pair. Either this episode is *about* the concept behind that word—the term is elite in this document—or the word just drifted past. You can't observe that switch directly; you only observe raw term count, a noisy symptom of whether the switch is on.



Corn

Walk me through how a binary switch turns into a curve, because if a document is either about archery or it isn't, why count past one?



Herman

Because one occurrence is weak evidence! Anyone can say "archery" once in a metaphor. Say it twice, my confidence that the episode is in the elite state jumps dramatically. Say it five times, I'm pretty convinced. Once you've said it twenty times, I'm already ninety-nine percent sure. Hearing it the twenty-first or fiftieth time gives me almost zero new information. Raw frequency is just noisy variation inside the elite state at that point.



Corn

So the tenth leaf proves you're in a tree, and the five hundredth leaf is just showing off.



Herman

Mathematically, that forces saturation. As term frequency goes toward infinity, a single word's contribution hits a hard asymptotic ceiling—which in Robertson's model is literally what the weight would be if you could see the hidden eliteness switch directly. No single query word, no matter how often it's repeated, can run away with the total score and drown out the other query words.



Corn

Okay, I'm bracing myself for some gorgeous twenty-term Poisson integral to get that curve, because I know that look on your face when probability papers come out.



Herman

Nope! This is my single favorite confession in the two thousand and nine monograph. They start with Harter's two-Poisson model from the seventies, where elite documents generate words at one rate and non-elite documents at a lower rate. When you write down the exact probability formula from two Poissons, it turns into an ugly, multi-parameter monster you cannot sensibly estimate on real text. So Robertson and Zaragoza literally write, "lacking an appropriate generative corpus model from which to derive a convenient formula, the authors of BM25 decided to fit a simple parametric curve to this shape."



Corn

Wait. They eyeballed it?



Herman

They picked the simplest algebra on earth that starts at zero, rises fast, bends over, and flattens at one: term frequency divided by k plus term frequency. Every search engine on the planet—Elasticsearch, Lucene, Tantivy, Postgres extensions—is running a hand-fitted curve in its inner loop because Robertson and Steve Walker on the Okapi team looked at a messy two-Poisson equation and said, tf over k plus tf has the right bend, ship it.



Corn

I have enormous respect for anyone who replaces a PhD thesis with division. What is k doing in that fraction, practically?



Herman

K, written as k-one in the full formula, controls how fast the curve flattens out—how skeptical you are of repetition. Typical textbook defaults in Manning, Raghavan, and Schütze sit between one point two and two point zero. Take k-one equals one point two. On the very first hit, one divided by two point two gets you almost halfway to the maximum score! By five mentions, you're above eighty percent of the ceiling. Crank k-one up to ten, and extra mentions keep adding weight much longer. Set k-one near zero, and counts above one barely matter—you're back to presence or absence. At the other extreme, if k-one goes to infinity, the curve never bends and you collapse straight back into linear TF-IDF, the degenerate case where repetition never gets stale.



Corn

Which brings a glaring problem to our archive even with the curve bending. Take the query "weird prompt." Or "honestly," which you said four times before breakfast. Every episode hits the saturation ceiling on those words immediately.



Herman

If every word had the same ceiling, "the" and "prompt" would dominate every multi-word search. That's where the second primitive sets the height of each word's ceiling: inverse document frequency, or IDF. Take the word "prompt" on a show called My Weird Prompts.



Corn

Worth zero. Less than zero, ideally—matching on "prompts" in "weird prompts battery chemistry" is just wasting electricity.



Herman

Practically zero is what IDF gives it! Unlike the saturation curve, IDF drops straight out of the Binary Independence Model—the Robertson-Spärck Jones weight, building on Karen Spärck Jones's nineteen seventy-two intuition. You take N, total episodes in our archive, about six thousand seven hundred, and n-sub-i, the number of distinct episodes containing that term at least once.



Corn

And take the log of the ratio.



Herman

The plain version is minus the logarithm of n-sub-i over N. Through information theory, n-sub-i over N is the probability a random episode contains your word, and minus log of that is the Shannon surprise—the exact information in bits of learning an episode contains it. If "prompt" is in every single episode, surprise is zero. BM25 itself uses the Robertson-Spärck Jones form: the log of N minus n-sub-i plus a half, over n-sub-i plus a half. The halves are smoothing so nothing blows up, and that version actually goes *negative* once a word is in more than half the corpus, so practical engines clamp it or add one inside the log to keep common words from punishing a document. Meanwhile, if "Okapi" appears in only two episodes, the surprise is massive. IDF multiplies the saturated term-frequency curve: TF decides how close you get to the ceiling; IDF decides whether that ceiling is ten stories high or two millimeters off the floor.



Corn

There's a trap in that rarity logic, though. We have ASR transcripts. Remember our episode on benchmarking custom ASR tools beyond word error rate, with all the mangled spellings speech-to-text spits out on proper nouns? A freak ASR hallucination appears in one episode out of six thousand seven hundred. Maximum rarity, maximum Shannon surprise, completely useless.



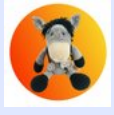
Herman

That tension is real: rarity and usefulness aren't the same thing. A word in two documents out of six thousand is phenomenal, high-precision evidence *when* it matches, though across thousands of searches ultra-rare words barely move overall system metrics because they touch almost no documents. Mid-rarity words are the system workhorses—yet for a human typing a three-word query, having that one needle-in-a-haystack word outweigh filler words ten-to-one is what makes keyword search feel magical. Computationally, it also lets engines skip giant posting lists for common words when their maximum possible contribution can't crack the current top ten.



Corn

Alright, saturated term counts, scaled by rarity. Now tell me why length ruins this, because I don't trust how a formula built for newspaper clippings judges a sloth and a donkey talking for forty-five minutes.



Herman

Document length is the sneakiest, most philosophical part of the equation. Imagine a ten-minute episode, fifteen hundred words, and a ninety-minute infrastructure deep dive, fifteen thousand words. Both mention "transformer" six times.



Corn

In the ten-minute episode, six mentions means half the show was about transformers. In the ninety-minute one, six mentions could be an intro tangent about power grids before an hour and twenty minutes on submarines. Penalise the long one.



Herman

You just stated what Robertson and Zaragoza call the **verbosity hypothesis**. Some speakers use ten times more words to say the same thing, or leave the mic open so random words pile up by chance. Under verbosity, six hits in fifteen thousand words is way weaker than six in fifteen hundred, so you divide term frequency directly by document length. Full normalisation. Now let me argue the opposite.



Corn

You're going to say it's ninety minutes because we covered ten subtopics instead of one. Power transformers, copper supply chains, grid inertia...



Herman

The **scope hypothesis**! Sometimes an author glued ten short documents end to end into one comprehensive piece. Suppose that ninety-minute episode has a self-contained ten-minute segment on transformers with those exact six mentions, plus eighty minutes of related engineering. Dividing its term frequency by ten brutally punishes comprehensive episodes and steers listeners to a stub!



Corn

Those two hypotheses demand opposite math. Verbosity says divide by length; scope says leave length alone.



Herman

And raw word counts can't tell whether an episode is long because Herman rambled or because the subject had eight moving parts. Real collections mix both, so BM25 compromises with soft normalisation. It computes a length factor, capital B: one minus little b, plus little b times the episode's length divided by average episode length across the archive.



Corn

Walk that extreme to extreme before the letters blur on audio. Little b goes from zero to one?



Herman

Zero to one. Set little b to zero, capital B equals one everywhere, length normalisation switches off—pure scope hypothesis, the old BM15. Set little b to one, capital B is purely episode length over average length, full proportional penalty—pure verbosity, BM11. And the universal textbook default shipping in almost every engine out of the box? Little b equals zero point seven five.



Corn

Quietly claiming that when a document is long, three quarters of the time the writer is rambling, and one quarter of the time they have more to say. Whoever picked zero point seven five had clearly read academic literature.



Herman

For nineties TREC newswire and government registers, it wasn't crazy! Snap the three pieces together so you see the whole machine at once. For each query word, BM25 multiplies its IDF rarity score by the saturating fraction: term frequency times k-one plus one on top, divided by term frequency plus k-one times capital B on the bottom. Notice length doesn't divide the final score; it multiplies k-one in the denominator. A long episode gets a bigger effective k-one, climbing the saturation curve slower and needing more mentions to prove eliteness; a tiny episode gets a smaller effective k-one, shooting toward the ceiling on one or two hits. Sum across query words, and that's your ranking. Before we see how it holds up in the wild, Hilbert has his segment.



Hilbert

Medieval Pottage Facts. The poor ate the offal and cheap cuts of slaughtered animals. The feet, the heads and the entrails. Someone got the feet. Someone always gets the feet.



Corn

Well, at least pottage didn't pretend to be twenty-five percent scope. Back to the eight-hundred-pound gorilla in Daniel's prompt, Herman: this math is a logarithm, an addition, and one division per query term, tuned on nineties newspapers. Why hasn't thirty years of machine learning buried it?



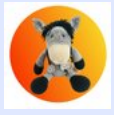
Herman

First, two knobs you barely have to touch. Robertson and Zaragoza note BM25 is shockingly insensitive to small shifts—whether k-one is one point two or one point six, whether length is characters, raw tokens, or unique words, top results barely budge. Meanwhile, try training a fifty-parameter ranking model: metrics like NDCG or Recall at twenty depend on discrete rank positions. Swap rank four and five, the metric jumps in a jagged stair-step; nudge a score without swapping ranks, zero change. No smooth gradients, just plateaus and local traps requiring expensive human relevance labels that go stale on new corpora, burning more compute to optimize than to run the engine!



Corn

While BM25 shows up day one with zero GPUs, guesses one point two and zero point seven five, and clocks out at five.



Herman

And look at the last six months of benchmarks, because people assume embeddings made BM25 just the cheap fallback. In May, Ke and colleagues evaluated software engineering RAG pipelines across twenty-one models and tasks—code generation, repair, summarization—finding BM25 stayed fiercely competitive and retriever choice often swung output quality more than swapping the generator LLM. In August, Cengiz tested Polish statutory law on bar-exam questions: BM25 put the exact right article at rank one roughly sixty-two percent of the time, beating dense neural retrieval by over nine points! And literally this week in November, Pickartz, Schaer, and Ströbel studied Heinrich Bullinger's sixteenth-century Reformation correspondence: on explicit patristic citations, BM25 hit ninety-four percent recall at twenty, while the best fine-tuned Latin bi-encoder maxed out at eighty-two percent.



Corn

Sixteenth-century Latin letters, software repos, Polish bar exams... what's the common thread where dumb word-counting humiliates neural networks?



Herman

Exact tokens carrying non-negotiable identity! A Python function name, a statute number, an explicit citation to Augustine, a part number, a surname. Dense embeddings squish five hundred words into one fixed-size vector of fuzzy semantic "vibe," blurring rare, crisp strings into soup. BM25 blurs nothing: if a string is in two documents out of ten thousand, IDF hands it a megaphone.



Corn

So the math was never the weak point. Saturation works, IDF works. What breaks is the human typing one word and the author writing another.



Herman

Bingo: **vocabulary mismatch**. Which brings the hammer down on the back half of Daniel's question: what happens when we point naked BM25—no helper, no synonyms—at six thousand seven hundred episodes of My Weird Prompts?



Corn

Spoken conversation is vocabulary-mismatch hell. In a blog post, Daniel picks "ASR," defines it once, and sticks to it. On air? One episode we say "ASR," another Hannah's setup comes up and we say "offline transcription," five minutes later you say "speech-to-text," and in our fine-tuning episode, half the transcript says "fine-tuning" hyphenated and half two separate words depending on the transcriber. To pure BM25, "ASR" and "speech recognition" share zero credit.



Herman

Worse on implicit concepts! In that Bullinger study where BM25 hit ninety-four percent on explicit citations, they also tested *implicit* references—paraphrasing a church father without naming him or quoting exact Latin. BM25 and dense retrieval recovered nearly disjoint sets! Papantoniou's Greek book-publisher study in July showed the same split: BM25 wins named entities and collapses on concept questions like, "that episode where Corn complained about devices built for human thumbs." If you actually said "claws on capacitive glass," BM25 returns zero.



Corn

Capacitive glass is a personal attack on tree-dwelling mammals and I stand by whichever words they were. What about the flip side, where the *exact same word* means two different things across six thousand seven hundred episodes?



Herman

Polysemy and name collisions! Think of "agent," "kernel," "mercury," or surnames like "Smith" or "Chen." At the start of November, Ning, Li, and Kong tested cross-topic name collisions across five hundred and sixty-four thousand arXiv AI records: under title-based BM25, shared names suffered nearly a twenty-six point drop in identification accuracy!



Herman (continued)

BM25 has no sense of neighborhood—search "Mercury" for planetary geology, and outboard boat motors or dental fillings get the exact same IDF boost unless you add a disambiguating word.



Corn

What about disfluency? You get excited and say, "Wait, let's look at the buffer, the ring buffer, the actual buffer size," tripling "buffer" in four seconds because your mouth outran your hooves. Does saturation save us?



Herman

My hooves stay politely under the desk! Nobody has published a clean BM25 benchmark specifically on disfluent podcast transcripts—the literature is all legal codes, medical abstracts, financial filings, and faculty pages. Reasoning from the formula, though, saturation is our best friend here: k-one bends hard after two or three hits, so "buffer, buffer, buffer" nudges the score from seventy to eighty percent of the ceiling instead of tripling it. Where speech *really* breaks BM25 is length normalisation!



Corn

Zero point seven five on our episodes!



Herman

Look at our archive: ten-minute SITREPs, twenty-five-minute episodes, fifty-minute deep dives. Why is a fifty-minute episode twice as long as a twenty-five-minute one? Not verbosity—you speak at the geological pace of continental drift regardless of runtime!



Corn

Consistency is a virtue.



Herman

It's twice as long because of **scope**—twice as many sub-arguments, tangents, and examples. Yet default b equals zero point seven five treats a fifty-minute deep dive as mostly bloat! Worse, Lv and Zhai discovered a mathematical bug in standard BM25 back in two thousand and eleven, inspiring BM25-plus: the length penalty isn't lower-bounded. If a long episode mentions a query word once or twice in a genuine two-minute segment, standard BM25 divides its contribution down so brutally that it scores barely higher than a tiny episode *missing that query word entirely* on a multi-word query! BM25-plus bolts on a floor constant, delta equals one point zero, just to guarantee matches in long documents a minimum pulse.



Corn

And where in the two thousand and nine monograph did Robertson prove k-one at one point two and b at zero point seven five were optimal?



Herman

He didn't! They're sensible defaults that fell out of nineties TREC runs on news articles, not a proof of anything. Plus, flat BM25 treats every word identically—episode title, Daniel's prompt, forty-minute monologue, or saying "Spotify" in every sign-off. That's why Robertson and Zaragoza built BM25F, giving titles, descriptions, and speaker turns separate weights and length normalisations. Still, if we dump raw transcripts into flat BM25, how low would b have to go before it stops penalising deep dives?



Hilbert

Zero point two, and it still ranks the repair supplement below the warranty card.



Corn

Zero point two is suspiciously specific, Hilbert.



Hilbert

My nephew Eli bought a nineteen-eighty-seven Mori Seiki lathe in October and put forty-one scanned shop binders in a SQLite table on a ThinkPad in his hallway in Petah Tikva. Left default b at zero point seven five because a tutorial told him to. Type "spindle bearing preload" and it gives you the two-page crate safety leaflet because the words appear twice in ninety words. The four-hundred-page overhaul binder with the actual



Hilbert (continued)

torque table comes up nineteenth. We've been arguing on WhatsApp since November fourth—he keeps lowering b by a tenth instead of fixing how he split the pages. Herman spent twelve minutes calling it the scope hypothesis when it's just one binder with nineteen machines in the back.



Herman

That's literally the textbook extreme of the scope hypothesis! Nineteen machines concatenated into one four-hundred-page binder share one huge denominator, crushing "spindle bearing preload" even at zero point two. Split the binder on chapter headers before indexing, and b wouldn't fight nineteen machines at once.



Hilbert

He won't split the PDFs because he wants row numbers matching the brass tags he riveted to the binders. The nineteen-eighty-seven overhaul binder isn't even in his flat anyway—his foreman Gideon took it to Modiin two weeks ago to trace a wiring diagram. Though what Corn said about spaces versus hyphens made me check Eli's query log ten minutes ago: half the missing torque tables say "pre-load" with a dash from an electric typewriter in Nagoya, so lowering b until Christmas wouldn't find them anyway.



Corn

He riveted brass database row tags onto paper shop binders?



Hilbert

Brass. Eli is twenty-six and very neat, Gideon is sixty-four and actually fixes the lathe, so nobody has turned a part since October.



Corn

At least when Gideon leaves the binder on his kitchen table, you aren't penalized for how thick the rest of the kitchen is.



Herman

Eli's binder exposes something crucial about our archive before we touch BM25's knobs: an episode doesn't have to be one giant index document punished by total runtime. Chunking into five-minute windows or speaker turns changes what "document length" even means, and every average-length number in the formula moves with it.



Corn

So you make structural bets the second you decide what counts as a document.



Herman

Before a single score is computed. And that's BM25 in one sentence: it is exactly as good as the words that happen to be on the page, counted honestly, and not one bit better.



Corn

Leave it to a nineties counting formula to outlive everybody. Here's our open question now, though: whether it's "pre-load" versus "preload," or "speech recognition" versus "ASR," pure BM25 hits a wall the instant two people spell the same thing differently. Before jumping to big neural embeddings, what sits between counting words and understanding them?



Herman

That is where part three of Search From Scratch picks right up next time.



Corn

Bring the tally stick.