FRED TALKS

7th April 2026

CONTENTS

[Daily article] April 7: Interstate 205 (Oregon–Washington)
ENGLISH WIKIPEDIA ARTICLE OF THE DAY · 309 WORDS

The AI Great Leap Forward
HAN, NOT SOLO · 2188 WORDS

The Rise of Transparency
ALLEN PIKE · 1316 WORDS

Session vs Query based search evals
DOUG TURNBULL · 1497 WORDS

[Daily article] April 7: Interstate 205 (Oregon–Washington)

ENGLISH WIKIPEDIA ARTICLE OF THE DAY · 07 APR 2026 · SOURCE

Interstate 205 (I-205) is an Interstate Highway in the Portland metropolitan area of Oregon and Washington, United States. The north–south freeway is 37 miles (60 km) long and serves as a bypass route for I-5 east of Portland. Such a highway was conceived in a 1943 plan for the area, and in the 1950s was included in preliminary plans for the Interstate Highway System. Construction began in 1967 with work on the Abernethy Bridge over the Willamette River, which opened in 1970. By 1972, I-205 was extended west to Tualatin and north to Gladstone, but the Portland section was delayed by political opposition until 1977. The Glenn L. Jackson Memorial Bridge (pictured), spanning the Columbia River between Portland and Vancouver, opened on December 15, 1982. The remaining 6.6 miles (10.6 km) in Portland opened on March 8, 1983.

> From Oregon City to Vancouver, the corridor is paralleled by a multi-use bicycle and pedestrian trail, as well as portions of the MAX Light Rail system. (Full article...).

Read more: https://en.wikipedia.org/wiki/Interstate_205_%28Oregon%E2%80%93Washington%29

Today's selected anniversaries:

1926:

Italian dictator Benito Mussolini survived an assassination attempt by Irishwoman Violet Gibson. https://en.wikipedia.org/wiki/Violet_Gibson

1994:

A FedEx employee tried to hijack Federal Express Flight 705 in a failed suicide attempt. https://en.wikipedia.org/wiki/Federal_Express_Flight_705

2001:

NASA's 2001 Mars Odyssey (artist's conception pictured), the longest-surviving continually active spacecraft in orbit around a planet other than Earth, launched from Cape Canaveral. https://en.wikipedia.org/wiki/2001_Mars_Odyssey

Wiktionary's word of the day:

suffering Moses: 1. (chiefly US, dated) An exclamation of dismay, irritation, or surprise. 2. About Word of the Day 3. Nominate a word 4. Leave feedback https://en.wiktionary.org/wiki/suffering_Moses

---

Wikiquote quote of the day:

Every great and original writer, in proportion as he is great and original, must himself create the taste by which he is to be relished. --William Wordsworth https://en.wikiquote.org/wiki/William_Wordsworth

---

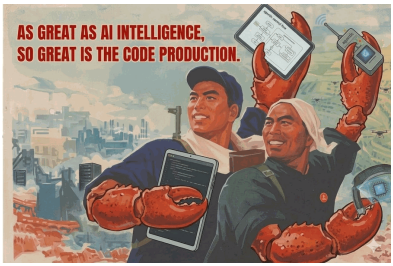
The AI Great Leap Forward

HAN, NOT SOLO · 05 APR 2026 · SOURCE

In 1958, Mao ordered every village in China to produce steel. Farmers melted down their cooking pots in backyard furnaces and reported spectacular numbers. The steel was useless. The crops rotted. Thirty million people starved.

In 2026, every other company is having top down mandate on AI transformation.

Same energy.



Backyard Furnaces

The rallying cry of the Great Leap Forward was — surpass England, catch up to America. Every province, every village, every household was expected to close the gap with industrialized Western nations by sheer force of will. Peasants who had never seen a factory were handed quotas for steel production. If enough people smelt enough iron, China becomes an industrial power overnight. Expertise was irrelevant. Conviction was sufficient.

The mandate today is identical, just swap the nouns. Every company, every function, every individual contributor is expected to close the AI gap. Ship AI features. Build agents. Automate workflows. That nobody on the team has ever trained a model, designed an evaluation system, or debugged a retrieval system is beside the point. Conviction is sufficient.

So everyone builds. PMs build AI dashboards. Marketing builds AI content generators. Sales ops builds AI lead scorers. Software engineers are building AI and data solutions that look pixel-perfect and function terribly. The UI is clean. The API is RESTful. The architecture diagram is beautiful. The outputs are wrong. Nobody checks because nobody on the team knows what correct outputs look like. They've never looked at the data. They've never computed a baseline.

Backyard Furnaces

Entire departments are stitching together n8n workflows and calling it AI — dozens of automated chains firing prompts into models, zero evaluation on any of them. These tools are merchants of complexity: they sell visual simplicity while generating spaghetti underneath. A drag-and-drop canvas makes it trivially easy to chain ten LLM calls together and impossibly hard to debug why the eighth one hallucinates on Tuesdays. The people building these workflows have never designed an evaluation pipeline, never measured model drift, never A/B tested a prompt. They don't need to — the canvas looks clean, the arrows point forward, the green checkmarks fire. The complexity isn't avoided. It's hidden behind a GUI where nobody with ML expertise will ever look.

The backyard steel of 1958 looked like steel. It was not steel. Today's backyard AI looks like AI. It is not AI. A TypeScript workflow with hardcoded if-else branches is not an agent. A prompt template behind a REST endpoint is not a model. Calling these things AI is like calling pig iron from a backyard furnace high-grade steel. It satisfies the reporting requirement. It fails every real-world test.

But the most dangerous furnace is the one that produces something *functional*. Teams are building demoware — pretty interfaces, working endpoints, impressive walkthroughs — with zero validation underneath. Some are in-housing SaaS products by vibe coding some frontend with coding agents: it runs, it has a dashboard, it cost a fraction of the vendor. Klarna announced in 2024 that it would replace Salesforce and other SaaS providers with internal AI-built solutions. What these replacements don't have is data infrastructure, error handling, monitoring, on-call support, security patching, or anyone who will maintain them after the builder gets promoted and moves on.

These apps will win awards at the next all-hands. In two years they'll be unmaintainable tech debt some poor soul inherits and rewrites from scratch. The furnace produced pig iron. Someone stamped "steel" on it. Now it's load-bearing.

Meanwhile, the actual product that customers pay for rots in the field. But hey, . The AI adoption dashboard is green.

Reporting Grain Production to the Central Committee

During the Great Leap Forward, provinces competed to report the most spectacular grain yields. Hubei reported 10,000 jin per mu. Guangdong said 50,000. Some counties claimed over 100,000 — physically impossible numbers, rice plants supposedly so dense that children could stand on top of them. Officials staged photographs. Everyone knew the numbers were fake. Everyone reported them anyway, because the alternative was being labeled a saboteur. The central government, delighted by the bounty, increased grain requisitions based on the reported yields. Farmers starved eating the difference between the real number and the fantasy.

You've seen this meeting.

One team reports their AI copilot "reduced development time by 40%." The next team, not to be outdone, reports 60%. A third claims their AI agent "automated 80% of analyst workflows." Nobody asks how these were measured. Nobody checks the methodology. Nobody points out that the team claiming 80% automation still has the same headcount doing the same work. The numbers go into a slide deck. The slide deck goes to the board. The board is delighted. The board increases investment.

Reporting Grain Production to the Central Committee

Then someone — there's always someone — builds a leaderboard tracking how many prompts you wrote this week, how much of your code is AI-generated, your ranking versus your team, versus your org, versus the entire company. One day your company announces: stop everything, it's AI Week. Build something with AI. Show what you've got. You think you're done after the hackathon? No no no. Now you have to *promote* it. Daily posts: look what I built, here's how many agents I used, here's how many skills I shipped. Pull in teammates. Pull in strangers. Ask for feedback. "Humbly."

Your AI usage is now a KPI. You are being evaluated on how much grain you reported, not how much grain you grew. This is Goodhart's Law at organizational scale: when a measure becomes a target, it ceases to be a good measure. The metric was supposed to track whether AI is making the company better. Instead, the entire company is now optimizing to make the metric look better. The beatings will continue until adoption improves.

Killing the Sparrows

The Great Leap Forward's most tragicomic chapter was the (Eliminate Four Pests Campaign). Mao declared sparrows an enemy of the state — they ate grain seeds, so killing them would increase harvests. The entire country mobilized. Citizens banged pots and pans to keep sparrows airborne until they dropped dead from exhaustion. Children climbed trees to smash nests. Villages competed for the highest kill count. It worked. They nearly eradicated sparrows.

Then the locusts came.

Sparrows ate locusts. Without sparrows, locust populations exploded. The swarms devoured far more grain than the sparrows ever did. The campaign to save the harvest destroyed it. Mao quietly replaced sparrows with bedbugs on the official pest list and never spoke of it again.

Every AI Great Leap Forward has its sparrow campaign.

Middle managers are the sparrows. They're declared pests — too many layers, too slow, too expensive. Flatten the org! Move faster! Let AI handle coordination! So companies eliminate M1s, turn managers into tech leads running pods, and let the teams self-organize with AI tools.

Killing the Sparrows

Then the locusts come. Those middle managers held institutional knowledge — which customer had the weird integration, why the data model had that inexplicable column, the undocumented business rule that kept compliance from flagging every third transaction. That context lived in their heads. Now they're gone, and the AI system they were replaced with needs exactly that context to function.

QA is a sparrow too. "AI writes the tests now." So you cut QA. The AI writes tests that validate its own assumptions — a machine checking its own homework. Senior engineers who mentored juniors? Sparrows. Documentation writers? Sparrows. The ops team that knew how to restart the weird legacy service at 2 AM? Definitely sparrows.

Each elimination looks rational in isolation. The second-order effects arrive six months later, and by then nobody connects the locust swarm to the dead sparrows.

Let a Hundred Skills Bloom

In 1956, Mao launched the (Hundred Flowers Campaign): "Let a hundred flowers bloom, let a hundred schools of thought contend." Speak freely. Share your honest criticisms. The Party wants to hear your real thoughts.

Intellectuals took the bait. They spoke openly.

Then came the (Anti-Rightist Campaign). Everyone who had spoken honestly was identified, labeled, and purged. The Hundred Flowers was a trap — an efficient mechanism for surfacing exactly who knew what, then eliminating them. The lesson every survivor internalized: *never honestly reveal what you know, because it will be used against you.*

Now Meta and a growing list of companies have launched their own Hundred Flowers. The mandate: every employee must build "agent skills" — distill your subject matter expertise into structured prompts and workflows that AI agents can execute. Or even worse, build "agents" using some drag and drop legacy tech that never worked and had already been given up by the leading edge labs back in 2024. Encode your judgment. Document your decision-making. Make yourself legible to the machine.

The stated goal is **distilling** your subject matter expertise. Turn the expert's craft into the organization's asset. What leadership actually wants is to convert individual human capital into organizational capital that survives any single employee's departure.

Employees see the game immediately. If I distill my ten years of domain expertise into a skill that any junior can invoke with a prompt, I have just automated my own replacement. The knowledge that makes me the critical node — the person they call at 2 AM, the one who knows *why* the model does that weird thing for Brazilian entities — is my moat. You're asking me to drain it.

So they adapt to build anti-distillation agent skills, just as the intellectuals adapted after the Anti-Rightist trap.

We are already seeing agent skills built specifically for job security. The *performative skill* looks comprehensive and demos well but omits the 20% of edge-case knowledge that makes it work in production — you are now *more* indispensable, not less. The *poison pill* encodes expertise faithfully but with subtle dependencies on context only you hold — internal wikis you maintain, terminology you coined, data pipelines you own — so removing you causes outputs to drift quietly until someone says "we need to bring them back on this." The *complexity moat* makes the skill so architecturally entangled with your other work that extracting your knowledge is harder than keeping you around. You are now a load-bearing wall disguised as a decoration.

The campaign designed to reduce organizational dependence on individual experts has now created experts who are *strategically indispensable* — not because of what they know, but because of how they've booby-trapped the system to need them. The flowers bloomed. They're full of thorns.

Meanwhile, the "everyone builds with AI" mandate has turned into a hunger game of scope creep. Engineers use AI to generate designs and ship prototypes without waiting for the design team. PMs use AI to write code and spin up dashboards without filing engineering tickets. Designers use AI to build product specs and run user research without looping in product. Everyone is expanding into everyone else's territory — not because they're better at it, but because AI makes it *possible* and the mandate makes it *rewarded*. The org chart says

collaboration; the incentive structure says land grab. What looks like productivity gains is actually a war of all against all, where every function is simultaneously trying to prove it can absorb the others before the others absorb it.

Engineering, PM, and Design scope creep

The Famine Comes Later

The Great Leap Forward's famine didn't arrive immediately. For a while, the numbers looked spectacular. Every province reported record harvests. Leadership was pleased. The requisitions increased.

The famine came when the real grain ran out but the reported grain kept flowing upward.

We're still in the reporting phase. The dashboards are green. Adoption is up and to the right. Every team reports productivity gains that, if summed across the company, would imply engineers are shipping at 300% efficiency while somehow still missing the same deadlines.

Underneath the metrics, it's a race to the bottom. One person builds a skill, so someone else builds a better one. One person demos a prototype, so someone else benchmarks it. Everyone competing to prove, more thoroughly than the next person, that their own role is replaceable. All accelerating. All sinking.

The sparrows are dead. The locusts haven't arrived yet. The flowers bloomed full of poison pills. The furnaces produced pig iron stamped as steel that's now load-bearing. The grain numbers look fantastic.

But it's fine. We're surpassing and catching up.

Oh, and Klarna? The company that loudly announced it would replace Salesforce with internal AI solutions? They quietly replaced Salesforce with another SaaS vendor instead. The backyard furnace couldn't produce real steel. They bought it from a different mill.

The question nobody's asking: what did any of this actually produce?

The answer, when it arrives, will be awkward.

References

- Kafka, P. (2026). Meta's AI week shows how every company is pushing employees to use AI. Business Insider. https://www.businessinsider.com/meta-ai-week-employee-training-claude-agents-vibe-coding-2026-3

- Blum, S. (2024). Klarna Plans to Shut Down SaaS Providers and Replace Them With AI. Inc. https://www.inc.com/sam-blum/klarna-plans-to-shut-down-saas-providers-and-replace-them-with-ai.html

- CX Today. (2025). Klarna Didn't Replace Salesforce — It Replaced Them With Alternative SaaS Apps. https://www.cxtoday.com/crm/klarna-didnt-replace-salesforce-it-replaced-them-with-alternative-saas-apps/


```
@article{
    leehanchung,
    author = {Lee, Hanchung},
    title = {The AI Great Leap Forward},
    year = {2026},
    month = {04},
    day = {05},
    howpublished = {\url{https://leehanchung.github.io}},
    url = {https://leehanchung.github.io/blogs/2026/04/05/the-ai-grea
}
```

The Rise of Transparency

ALLEN PIKE · 01 APR 2026 · SOURCE

Small companies are, by default, very transparent. When there are 4 people working in a room, you have a direct line of sight on what everybody else is doing, and why. Your docs, Slack channels, and repositories are open to everybody. When the CEO has an epiphany that changes everything, you all know right away – probably because you were at lunch together when it happened.

Thus, startup founders will often get religion about transparency. “Our culture,” they’ll declare, “is to be radically transparent! Everything defaults to open. We hire adults, expect them to do great work, and give them the context they need.” Yay transparency!

And this works pretty well. Transparent orgs tend to delegate more effectively, have higher accountability, less politics, faster trust, and just plain ship more. Transparency helps bigger orgs adapt more quickly to the ground truth, responding to customer signals that execs might not be directly exposed to.

But, at a certain scale, radical transparency strains.

Some idle musing by the CEO sends a team off on an unimportant side quest. A well-justified compensation anomaly upsets a group who is missing background information. A 450-message Slack thread about bike shed paint color choices devolves into factions, hashtags, and philosophical arguments about the morality of taupe. #nevertaupe

And if you talk to people at a large yet highly transparent company, you’ll hear about the hazards of the relentless **firehose**. A thousand shared Slack channels, to start. But also a glut of docs – some critical, most unmaintained. Then there’s the meeting notes, meeting recordings, and meeting invites. Plus proposals, requests for comment, and requests to comment on your proposals’ comments’ resolutions. “So, you like information, eh? Well, have all the information in the world!” How do you make sense of all this?

While some people are tenaciously able to find, within this chaos, the important info they need to do great work, a lot of otherwise-capable people get easily distracted by information that just *might be* urgent, provocative, or even just… shiny.

Meanwhile, allowing everybody access to every historical doc is occasionally useful, but it also presents an ever-growing surface area for leaks and legal liability. Are you sure there isn't something highly sensitive or disagreeable in those 99,999 unmaintained Notion docs?

So, as companies grow, they tend to lock information down. Some – Netflix, Stripe, Shopify – do their best to keep as transparent as possible while still complying with necessary guardrails. Others – Apple, Palantir, Oracle – move toward a need-to-know basis, ensuring information flows top-down. With more control over information, it's easier to ensure that leaks or internal distractions don't derail your plans for surprising product launches and/or world domination.

Of course, every company's culture is forged by the market they operate in, but there's always some tradeoff here. And as companies grow, they tend to regress to a boring middle ground.

However. As with many tradeoffs, the balance has recently begun to shift.

Given this firehose, please assess my plan

Recently, we've seen a revolution in tools that can make better use of the firehose. Slack can now summarize your unread messages, albeit with mixed effectiveness. Tools like Glean and Unblocked can consider a mountain of your company's data and answer important questions about it, albeit limited to the data they can actually see. And large open companies like Shopify and Stripe have internal tools that let employees' agents query, analyze, and act on the copious data any given employee has access to – albeit with some sharp edges and exfiltration risks.

Just as LLMs are making the world's data more useful to the world, they're making companies' internal data more useful to employees.

Of course, this can be misused! In some companies we'll see further secrecy – I've heard of AI search tools and MCPs letting employees find accidentally-visible compensation data and other spicy docs that hadn't been audited. I've heard of support agents giving customers true-but-problematic information because they surfaced it with internal AI tooling without proper training.

But as we evolve past early growing pains, and into teams and processes fully making use of this stuff, the anecdata points toward this new tooling becoming a superpower. Agents' newfound ability to effectively query and reason about far more data than can fit into context is making the long tail of communications and docs much more useful for decision-making – but only when people have access to the relevant data.

Given that, **the maturation of AI tooling will motivate companies to become more transparent**.

In 2024, the cost of being internally secretive was meaningful but manageable. Although Apple keeping information need-to-know sometimes leads to waste, or important changes being slow to diffuse through layers of management, they've done, like, pretty well for themselves? With all the scrutiny from press, competitors, and regulators, you can see why they've kept it up.

But as all companies increasingly have tools that can assess, consider, analyze, and make use of all the business' communications and documents, what kinds of org are going to benefit most? Well, the ones that let their employees access more context.

Extremely transparent orgs like Zapier, GitLab, and PostHog that might have struggled to cope with their firehoses – and who often had gaps in the data due to untranscribed meetings and decisions – will increasingly be able to leverage it. Sure, not all of it, certainly not at first. (Some of it is just junk.) But increasingly more of it. And critically, it won't just be executives that will be able to attend to all this knowledge.

Were we ought to be

The frontend dev working on your internal admin dashboard should be flagged that the React upgrade issue they're battling right now was just solved by the customer-facing dev team. The intermediate developer who is incensed about a company-wide tech decision should be able to build their understanding of why it was made without booking a 1:1 with the responsible Principal Engineer. Your go-to-market team should be able to "see" through to the code, developers' conversations, and the recent decisions around a given feature, letting them give customers correct and timely information about what to actually expect from the product today.

And everybody in your company should, when it's useful, have key company-wide strategy docs available to their agents as they make plans and decisions. And then, when a new revelation motivates the exec team to improve those docs, then bam. All the product engineers' agents will take this new strategy into account right away. Anybody who's worked at a large company and/or used CLAUDE.md knows this won't be a silver bullet – deeply ingrained habits and momentum can not be simply prompted away. But as the tools and the data improve, the advantage will accumulate.

When we launched a realtime meeting agent last month, we expected to get feedback about its defaults being too open – currently, Cedarloop defaults to sharing its collaborative notes and tools with all attendees live. But instead, we've seen two diverging kinds of feedback: many of our users want the tool to be less visible to external guests and customers, but *more* open

internally within their companies. Which in retrospect makes a lot of sense: decisions and actions in your team's work are increasingly useful across your company, but your customers shouldn't need to worry about all that.

So long story short, more internal transparency is coming.

It will take some time. Apple isn't doomed, and just because Zapier and Shopify are already working that way doesn't mean they're going to instantly be turbo-boosted. But it seems a new era is coming, where siloed knowledge, information hoarding, and secrecy-by-default will become less tenable.

The firehose will evolve from a spicy distraction to a useful input to important work.

Session vs Query based search evals

DOUG TURNBULL · 30 MAR 2026 · SOURCE

To evaluate search, we typically build a judgment list. We transform clickstream data into evaluation data. This labels a result as relevant (or not) for a query. Let's walk through an example with movie `rocky` - first to see the classic method.

At some point in the past we observed this session:

Rank	Query	Document	Click?
1	rocky	Rocky	False
2	rocky	Creed	True
3	rocky	Alien	False

And some other time, for query Rocky, we observe another interaction:

Rank	Query	Document	Click?
1	rocky	Alien	False
2	rocky	Star Wars	False
3	rocky	Rocky	True

Adding up these two, we'd have:

Query	Document	Clicks	Impressions	CTR
rocky	Rocky	1	2	0.5
rocky	Creed	1	2	0.5
rocky	Alien	0	2	0
rocky	Star Wars	0	1	0

Now we can use that to put a number on whether we produce good results for `rocky` -

```
# Some offline experiment produces these results for rocky
q=rocky
1. Creed (0.5) <-- we label each row with its relevance for the query
2. Rocky (0.5)
3. Star Wars (0)
```

```
Here we could compute all kinds of statistics (ie [NDCG](https://www.

I call this model query based evaluation. There’s nothing wrong with

## **But there’s a different way - session based eval**

What if we just directly used the original session. We “replay the se

For example, evaluating against the first session, we get
```

q=rocky

1. Creed 👍
2. Rocky
3. Star Wars ```

We’ve placed the clicked item towards the top. The second session we get:

``` q=rocky

1. Creed
2. Rocky 👍
```

3. Star Wars ```

For the second session, the clicked result lives in the second position. Not as good.

When we average 1 / rank of the right answer (mean reciprocal rank). Here that'd be (0.5 + 1.0) / 2 = **0.75**

Essentially we decide there's no *one right answer* for the query. The question we answer: would we have satisfied some past user? Based on a single event of user searching + clicking, would we have done better by that user?

Here we've stuck to one query. But that's not how session based sampling would work. In reality, we'd first sample some N sessions regardless of query. We expect that N to fairly represent the user experience. It might have 5000 queries? 3000? It doesn't much matter.

Let's walk through why we would / would not take this approach

Advantage 1 - Improved sampling accuracy

Remember that when we ship this to production, the question won't be "how many queries did we improve?", it will be "did we improve most user's experience with search?". Users experience search in sessions, not in queries.

With query-based evaluation, good teams try to fix this. We reweigh each query to get back to how many sessions it represents. Very popular queries get higher weight proportional to how many sessions it represents. Tail queries much less weight.

Further with query-based evaluation, we sample at the wrong level. We put the cart before the horse. We don't try to get a representative sample of user sessions. We instead try to identify 1000 or so queries we want to fix.

Political polling serves as a useful analogy. Imagine you wanted to poll the electorate about an issue. What if you first identified the cities you wanted to poll before arriving at a conclusion. So you choose 1000 random cities. Ranging from big cities like New York to smaller towns like Chingoteague, Virginia. You poll these places, and then work backwards: weighing the New York result higher than the Chingoteague.

That's a bit backwards. Nobody polls like that - they just try to truly randomize the voters to get a picture of the overall electorate. Pollsters call this probability based sampling - every person in the population has an equal chance of being polled.

Session based evals look more like probability based sampling. Every user interaction has an equal probability of being sampled into evaluation. Query based evals look lumpier - like getting a view of specific targeted cities, then trying to work back to the overall population. Not impossible, but another source of error.

Advantage 2 - time-sensitive ranking features (ie in learning to rank)

Another advantage comes when we used this evaluation data as training data.

In a query-based eval set, we aggregate user sessions over a lengthy period of time. Maybe a month of user interactions. That gives our per-query labels more confidence.

However, the major downside comes in WHY users might click result - some condition that was only true that specific day they clicked the result. In e-commerce, for example, pricing can be dynamic. Some hour there may be a very specific reason the result appears ‘relevant’ - its temporarily on sale!

When we train a reranking model, the evaluation data we’re discussing becomes training data. With learning to rank, we learn why users prefer some results from the features (bm25 scores, embedding similarities, document popularity, product price, etc).

So when aggregating a months worth of query clicks into a label, we must also try to represent what features describe the query / document over that whole month. Averaging, for example, price would hide the reason one item got a tremendous amount of clicks over a short period of time. We can’t train the model to learn that sudden price drops cause more clicks.

With session based evals, however, our training examples come from a single point in time. We can capture what was true at that moment (price, etc). That makes a big difference in learning these patterns.

Disadvantage (or question?) - we still have biases

It’s important to note that clickstream biases DO NOT go away in session based evaluation. A click far down the search results page would, all things being equal, be rather surprising. If we saw that item clicked frequently farther. down, we might consider rewarding it more than an item clicked close to the top. And nothing here solves the echo-chamber effect we call presentation bias.

We likely need to construct a weight to each position directly. For example, computing expected clicks per position, then computing a neutral “probability of click at position” as a kind of prior. Our session might actually be then a set of labels that move labels more negative when they’re not clicked per position, and much higher when they do receive clicks. That might help us debias the data somewhat in a principled way.

In effect, we don’t end up counting every click as a 1. We count every click as some weight according to what position we observed it in (farther down, higher clicks).

Disadvantage 2 - debugging individual queries

Another challenge comes when trying to get to the bottom of a single query.

Return to our polling example. If we first polled many cities (Chingoteague VA, New York City,) then tried to aggregate to a whole, it might add a lot of noise to the picture of the entire electorate. However, it clearly would give us the advantage of understanding those specific cities. That’s the advantage of query-based evaluation: a debuggable per-query picture.

We could NOT for example, understand New York City just because our sample of 1000 voters happened to sample one NYC resident. In the same way, our sample of a query might just luck out on a single session of a popular query (ie `rambo` above).

In a sense, session based evaluation tries to use historical data to recreate an A/B test. Just like an A/B test, a single query may be too noisy to evaluate. Intentionally selecting for a population of queries, however, lets us get a deeper picture into patterns where search goes wrong.

Final thoughts -porque no los dos?

It’s worth noting that both solutions share another important bias - they try to use past data. Tomorrow, users may wake up, and decide other factors might be important to them. Some world event may suddenly drive interest in energy efficiency, for example.

That said, both systems derive from a set of sessions, and give you a different picture. Don’t treat either as obvious “truth” - just useful models. Reconstruct query-level evaluation when you want to debug (knowing you sacrifice time-sensitive features). Sample sessions when you want to approach a “simulated A/B test” - noting you’re sacrificing per-query debugability.

Like everything else, no model is accurate, some are useful.

-Doug

PS - hope to see everyone this week at Leonie Monigatti's talk on Context Engineering+Agentic Searchand my chat with Brian Pedersen on search+tech career FAQs

… and next Cheat at Search cohort Pricing goes up in 7 days