How to debug Retrieval for RAG

41 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=N9SZ9O2GREE
https://www.youtube.com/watch?v=N9SZ9o2gReE

SUMMARY

Doug Turnbull discusses the evaluation of retrieval-augmented generation (RAG) systems, emphasizing the importance of distinguishing between different types of evaluations. He outlines a framework for assessing retrieval quality, focusing on the role of retrieval in AI applications and the need for iterative improvements in search tools.

- *Evaluation in RAG involves two main types: precheck evaluations (similar to unit tests) and AB tests for user satisfaction.*
- *It's crucial to differentiate between research datasets and practical evaluations tailored to specific use cases.*
- *Traditional search evaluation metrics like DCG and NDCG help assess relevance, but RAG requires a different approach due to its complexity.*
- *A seesaw model of evaluation is proposed, where improvements in one area may temporarily lower performance in another, emphasizing iterative development.*
- *The importance of labeling search results and understanding user intent is highlighted, along with the need for a baseline search tool to start evaluations.*
- *Doug advocates for focusing on user experience and engagement rather than solely on business outcomes in RAG systems.*
- *The discussion includes the evolving landscape of search and retrieval technologies, including the integration of AI and agentic systems.*

So today we have Doug Turnbull. He is the OG of retrieval. He's been doing search way before search was cool, way before LLMs or anything. When we talk about rag, the R in rag is retrieval. Doug's been doing search longer than anyone that I know. Of course, there might be older people than me that there might know people are doing longer than that, but I can't think of anyone that you can learn from that's better at search than Doug. And today he's going to be talking about evaluating rag. I'll hand it over to Doug. Yeah, thank you.

So we're going to talk about evaluation. What is it? And in rag, what is that? And probably don't need to define evaluation too much for this crowd, but I might as well just say really what I'm talking about is evaluating a specific part of the stack. When I think about rag, I'm assuming some basic facts, right? That there is an agent, the agent has tools, the tools can call retrieval systems that try to return useful results to an agent. And the agent has various features like it's able to do reasoning, it's able to think about what it got back, it's able to adjust its path forward. So there's a lot of bells and whistles in there. Mostly what I care about is the retrieval part of this whole picture that frankly drives a lot of the quality of how the agent works.

The whole overall picture of like, hey, there's an AI application and does it work or not work? I think that's what you're all here to learn from Hamel and Shreya. And I want to differentiate between two types of evals and

this really in the search space trips people up. I can't tell you how many times I've worked with a team who's trying to label some results. And the implicit assumption when you talk to a manager is that the goal of those results is that labeling that you're doing is to like somehow predict that you're going to win an AB test. And I think it's important to decouple two types of evals that I'm going to talk about. The first type is just this what I call precheck evals, which is almost like a developer, how we think of unit tests. So like does it work as I expect? You're going to hear me slip into slightly, sometimes I'll go code switch I guess between like the search world with search bars and search results versus the rag world because I tend to live in both. But in the search world, you're always trying to get to an AB test, right? And it's often more useful with this precheck to just be like, hey, is the AB test even going to measure what we think it's going to measure, right? I heard this funny phrase when I was an early developer like works as coded. It's a little bit better than that, but it's sort of that notion. Like does this do what I expect it to do? Then there's a second type of evals, which is really your like AB test. And when people talk about evaluation, they'll sometimes mean this.

Your plane, your search system, your whatever is off, it's helping users, and you're trying to sort of read the tea leaves to see if those users are happy about the change. Whatever we're measuring, we need to think about the role of are we in this precheck world that's useful to us as developers to kind of make sure we're approaching what our expectations are met. Like the unit tests are passing, the integration tests are passing, or this other world which has a lot of domain expertise, may have a lot of data science depending on what you're doing. And I want to separate those somewhat. I want you to think about those differently. I want you to think that evals can serve different purposes along the stack. A lot of times with this comes up is users just doing surprising things in search engines and throwing you for a loop. Just don't confuse these types. And I'll talk more about this in a bit. So we want to distinguish between the kind of works as expected and shows user satisfaction.

That's a bit of a different thing. And the other big thing, this is a huge deal in search, is don't confuse a research data set, which is some generic objective like is in the static pristine set, it's supposedly comprehensive with all the use cases you're supposed to care about, it's probably for a generic domain. That is a very different thing than your evals that you are building. I don't want you to confuse those two because there's a lot of decisions you have to make in your own evals. I don't really get into it in this talk, but for things like if you're labeling search results, like what do you do when a search result comes back that's unlabeled? Something that can throw you for a loop. And in a lot of academic search data sets are like if you get an unlabeled search result back, just assume it's irrelevant. And they sort of just move on. But we can't do that in real life. We have to actually think about these problems and think about what we care about measuring. In this talk, I'm going to talk a little bit about general search evals just as like a baseline definition. Some of you may know this.

And then we're going to get into rag specifically. It's sort of like how I think about the rag slash agentic space when I'm thinking about retrieval problems there. So the typical search evaluation, there's some queries. We use the term document frequently even if the document is a product or the document is a job. It's like the search result. And there's some grade or label that says how relevant that document is for the query. And this is sort of canonical example that I carry with me all the time is just some movie search. You've got Rambo, you've got Rocky, and you can see Forrest Gump obviously is not relevant for Rocky or for Rambo. First Blood is obviously very relevant for Rambo. It helps us sort of evaluate search results. And you have you can ask yourself where these come from, clicks, humans, LLMs. If we were talking and doing one of my search trainings and we were

talking about traditional keyword search, we would spend probably half the training on how to devise these labels in some way, especially if you're going to then train a model on them. Because it's depending on there are companies, I know I worked with LexisNexis, they have an entire team that does click-based evals, they have an entire team that does human evals, and they were experimenting with LLM-based stuff. But and then crowdsourcing is sort of its own subspecialty of humans. It's complicated. But we're going to set that aside because honestly for a lot of the evaluations you do for rag, for the 80% getting started value, a lot of that complexity you don't need to think about it yet.

You may not need to think about it at all. But to get into the nuts and bolts of how do you would use this, you would have some search results. This is the thing we want to evaluate. Remember we're just talking about traditional search before we go to rag. It's not good that Forrest Gump came up higher ranked than First Blood or Rambo. So we take our labels. We have labels for all of these movies relative to the movie Rambo and we plop them next to the movies. We are going to do kind of a weighted average. Here we're doing a It's called a discount. Here we're just going to use one over the rank. So this means that the top results are more important than the second and and so on.

And then multiply that through sort of weighing the grade. 0.01 * 1 because that's a discount is 0.01. And then this is a grade one, but it gets taken down by half cuz it's down a bit farther in the position and so on. And we add those up and we should get a DCG. We call that DCG, which means discounted, just means weighted, cumulative, which means added together. Gain is cuz that's the thing we're adding. So this is just one of many.

There are dozens. I think at one point when I worked at Open Source Connections, I think we counted two dozen of these statistics that you can take basically a judgment list and come up with a number on how good your search results are. You may not hear about this statistic called NDCG, which really takes this DCG out of a total what whatever the ideal DCG is. So if you take the judgments and you sort them in some ideal order, you get ideal DCG, IDCG. And if you divide that, you get what's called NDCG. It's like sort of scales it from zero to one. So that's one way of thinking about search evaluation. And early in my career I would go around and sort of tell people, hey, you need to like figure build judgment lists and and think about these things. The main thing to think about if you are in this is think about do all of these factors that I'm going taking a judgment list and then coming up with a number on a query, do these actually matter for my problem? So you can imagine situations like this comes up in e-commerce all the time in a traditional search setting where you have like a grid of results. So you have four results at the top and then another four and so on. And actually something like NDCG doesn't make sense because there's not really a bias into how the user perceives the ranking. You can instead just literally take the average of the grades of the labels of the top four because usually that's above the fold. And that's often referred to as precision, just taking sort of portion of relevant results above the fold. So a lot of the how you choose these things and these numbers and use these statistics, it's very domain specific and it matters a lot to your specific end application. And you can imagine an LLM is going to have its own biases potentially in the context it's getting from your search results. I have a question and you may be getting to this. What's the best place to start?

Do you need to have labels? What if you don't have labels to begin with? Do you have any I'll get to that in a

second. But uh that's a great question because most people are not, the vast majority of teams building some rag thing are going to have any labels. It's actually maybe because of the teams I consult with, but say good proportion of them are somewhere along the way where they have some mature labeling system that seems to work for them. But yeah, I'll get to that. So here's something to think about.

Everything I just showed you, would this be something I described earlier as a precheck metric? Is this a precheck eval? One way this might be a precheck eval is if this is just developers labeling the results. There's really no wrong answer here. It's only like documenting your assumptions. Hamel and I are working on a project. We literally just made a change and we want to make sure that the five queries and the search results that we would expect to come up came up. We don't care or know, or we do care, but we don't really know if this is a thing that's going to be a big win in some AB test. We're just like tweaking and tuning search results, and we want to prove that we accomplished our task as developers. So, that's like more of a what I talk about as a pre-check. It's like, are we doing what we expect? Are we implementing the change we expect? Or are there like weird side effects and we change a bunch of other queries and that kind of thing.

The other universe this comes up is more of a proxy for what I called an in-flight eval. So, you're almost like a proxy for an AB test. Like, are we taking clickstream data and labeling results that get really good sales for this Rambo query and assuming that if we rank those higher, that we're going to have a better AB test. That level of like statistical modeling and data science, uh that that's complicated and you can imagine entire teams existing to try to produce a number of how {quote} {unquote} relevant this result is for a query. Trying to predict conversions, KPIs, and things that like actually matter to the business and users. And these are different tasks and there's no wrong answer. The main reason you'd want judgments that did this is maybe to start to like maybe train a model off this data or like a ranking model or something to try to promote things that you predict would get more conversions and that kind of thing. But, there's all kinds of problems and biases in there.

There's an echo chamber effect of users only clicking or or seeing or interacting with what they see and that kind of thing. So, there's no wrong answer and there's many ways that you can plug into this architecture to try to measure ranking. So, MRR is another statistic people may know about, which is basically how far down is a relevant result. It's basically the average of one It's like for a query, it's one over the rank of the relevant result. So, if the relevant result showed up in the second row, the RR reciprocal rank would be one over two. And that is usually used for in like question answering context where there's literally one answer to the question or in other contexts where it's like, I need to prove that the one answer is kind of closest to the top of the ranking.

Whereas, DCG, what it's doing is like, there are many potential gradations of right and wrong answer. Let me make sure that those show up somewhere in there and that I've gotten a good ranking. None of this, everything I've just described, is traditional search. RAG doesn't really feel this way. It's not like we're getting search user search results for two keyword queries and expecting users to interact with them. There's some major differences here. And if we walk through the use cases, to me, there's kind of a ladder of progression of complexity from like traditional what I would say traditional RAG where it's literally that almost looks like search results like this. I was learning about this. If you want a good podcast, there's a really good one called History of Industrial Revolutions podcast. It's not Revolutions, but Industrial Revolutions. And there's this fun

guy named John Ironmad Wilkinson who was just obsessed with iron. And I searched for him and I wanted a summary. And these results are almost search results.

There's almost 10 results and maybe I could interact with them and you'd get some data on like how I interact with them. You also have situations that are like synthesis. So, is there evidence that Family Guy was inspired by The Critic? This is all ChatGPT and here it's pulling from different sources to come up with >> [snorts] >> a perspective. And it's like summarizing that. This is like a ladder up in complexity. It's not like this prompt whereas before the you might take the prompt as the query directly and find some things. This is like a prompt that that gets broken down into many different searches that help ladder up to answer a question. And then the very top of the pyramid is literally an agent taking some action for you. Like, buy me affordable yet fashionable sunglasses.

And you trust the agent to do the right thing. So, this is your like literal open claw situation. So, this is somewhat befuddling to my innocent search brain that knows about 10 hot links and wants to put the relevant stuff towards the top. But, you can kind of start to build a framework for how people might think about this cuz you have this high degree like you can think about this as a pyramid. Basically have summarizing direct results where literally the results are interleaved up to a synthesizing step and then very top you kind of have some step where you're taking action for the user. Like, buying sunglasses for me. And as you move up the ladder, down here, the quality of the final result is almost entirely dominated by retrieval. As you go farther up the ladder, it's retrieval is one component of an agentic system that may or may not satisfy the user. So, at this top level of the ladder, it's really hard to do uh focus on like you should focus on retrieval quality, but the big question is more about the entire AI system and whether or not it's so solving users' problems. Whereas down at the bottom, we're much more focused on like it feels much more like a retrieval problem. The reason I do this is to kind of get a sense for how I as a retrieval person would think about evaluating the retrieval part of the system. At the top, the agents kind of might even participate in the retrieval evaluation under human supervision. And I want to make sure it's not quite like LLM as a judge, but it's almost like the real right answer is like whether or not the agent bought me good sunglasses. And then I might ladder go down into reasoning traces to try to figure out, is the search tool doing what I expect it to do? So, it's a bit of a different world. But, it's in service of the agent, not directly in service of the human. Whereas down at the bottom, it's more akin to like the search evals I've already talked about. It's literally it's the the results are more dominated by what the retrieval does with some like AI summarization in there. And then at the you can almost think about 10 search results is like, I'm the AI. I'm the person who has to figure all this out and summarize things. So, unlike traditional search, so traditional search we can like see a single two-word query a thousand times a day.

Prompts occur too infrequently to look at user clickstream data and somehow learn some patterns. We probably will never see this prompt again. That may not always be true. There's probably some repeated prompts, but we in search we say there's like a long tail. There is huge distribution of these queries. RAG users almost never click. You might be able to do some things and this isn't true of all RAG-like systems. You might be able to do some things that invite interaction with the results to kind of get clickstream based feedback on them. Like, highlight or click to read more or whatever. But, by default, you're sort of expecting a user to sit there and read some results. And I also want to say there are some systems that I consult with that are maybe somewhere between RAG and traditional search where literally it's a prompt and then 10 search results or some search results that a

user goes through and maybe interacts with in some way. So, an example of that might be describing the ideal candidate for a job. And there might be a version of that interface that looks somewhere between the RAG system that's chatty and a system that is traditional search to kind of take a prompt, show you search results, and let you chat to interact with those search results. And that is another place we might be able to use some degree of click analytics. And then the question becomes I I used this analogy before about the four results above the fold of a e-commerce site, is ranking the right thing to think about for RAG. Like, one thing we obviously care about is token efficiency. We care about with retrieval giving RAG the right search results and not a lot of irrelevant search results. I think early in my RAG consulting, a lot of gigs would be like, "Hey, we shoved a hundred results into the LLM. Now it's throwing out context. What do I do? How do I narrow this down?" So, it can be as much about token efficiency as anything else.

And what importantly, we care about the final task. Was the final task made easier or harder? Now, I'm going to get even more opinionated about if basically if I was parachuted on RAG team, your RAG team, any RAG team, what exactly I would do. So, the first thing if it was up to me and I was parachuted in to be god of RAG at whatever company, the first thing I would do is build a baseline tool. You can see it to hell with my consulting cuz I like pyramids, but the foundation is just a dumb baseline search tool. And I want to differentiate between what I'm talking about and what maybe a lot of teams end up doing. I often just am happy if specially for I work with teams that maybe already have a search system, but just taking the dumbest thing that could possibly work, putting it behind a tool an agent can call, and just getting it out there. By shipping it maybe not directly to users, but whatever the closest thing to prod you can, get it out there. BM25 is a type of search you may have heard of. It's just keyword search, which a lot of teams may already have. Where I see teams fall apart with RAG is they will spend six months building a giant scaling up a vector database, chunking their content. You might need to do chunking eventually, but starting out with this big waterfall project of we're going to scale out our vector database, they don't have a vector database, realizing they want to chunk everything into sections and then getting embeddings for all those chunks.

That is actually turns into a pretty high scale problem pretty quick. It's easy to get into tens of billions of embeddings that way. You don't need to do that. Don't get distracted by that sort of classic RAG architecture we talked about like two years ago. That's one strong opinion weakly held that I have about this stuff. So, start simple. And really what you want to do when you're working with an agent in retrieval, is you want to create a search tool with a promise. Here I've used this example where I'm going to use these examples. We're going to start looking at people search. So, these are two tools and they have different promises. People search, maybe there is we might say something a bit more about the purpose. Maybe it's a LinkedIn people search or something. We expect to take a name, a some profession, whatever that means and some other attributes to search for. That is a tool and a lot of times in these agentic frameworks like the docstring or even the function name and the parameter names all become part of the context, all become part of the prompt so the LLM and the agent know how to call it. Versus web search and web search sounds fairly open-ended, but it's actually its own kind of promise.

Like searching like Google where authoritative information comes up towards the top. Where it's sort of capturing the entire world's knowledge essentially that's that's up to date. That's a different kind of promise. The other thing you want to do with getting started is to think when you have a fairly open-ended tool, when

there's a pretty big gap between the thing you want to accomplish and the tools, like the tools are just open-ended, that's where skills are coming into play. For example, if we we skills also might apply to your rag and agentic search system. You might have something like, hey, to search people on the web be sure to put their name in quotes or the web search engine can focus on sites. So, when you're searching for people, like this is the people search but through the web search tool skill, when you search for people like LinkedIn, you can filter to that by doing site:linkedin.com.

That kind of thing. And this is uh part of extending the promise that these tools are making to the agent. So, with a promise defined, we have kind of a contract that we're starting to develop between the agent and what the search is able to do. So, the next step when we're thinking about evaluating getting started with rag evals is really answering the question are we fulfilling the promise? And that's where we start to get these kind of pre-flight evals that I think of as internal to a search team. I don't necessarily think of this as the are we going to win at the AB test or does this prove that search solves our users' problems? It's really is this fulfilling the promise that the search laid out. And this kind of gets to Hamel's question. I mean, the very first thing you should do is just start to label some results. Get started in a spreadsheet. And this literally could be the me and Hamel are working on a problem and we just want to get some sense of like, hey, here are the things that we want to come back for these queries. This is sort of like extending that people search.

So, there is a Doug Turnbull that works in search that is my doppelganger who is a professor at Ithaca College. He's a professor in information retrieval. And so, this is kind of just kind of a fun example. So, you search have these searches Doug Turnbull Ithaca College, maybe the Doug Turnbull from software Doug. Doug Turnbull is is irrelevant. And we start out with some basic of evals. Here I've changed instead of numbers, we're just doing thumbs up, thumbs down. And it's important to literally just build a spreadsheet. And this is helping us try to figure out does our tool work as expected, not will this please the user.

The other thing to keep in mind when you're working this way, your goal is actually not to have a graph that goes up and to the right. Your goal is to have a seesaw. You have some initial evaluations, you make them better, you decide that you want to tackle a new use case with your retrieval tool to make it better at something. And there's always something that your retrieval tool is not good at. So, here we've taken that initial set of labeled results and we've gotten to a good point and then we add a new use case. And that's going to make our whatever metric we're look we're using to measure our retrieval quality go down. So, we have some new evals that cover that new use case. And we make that a little bit better. And now we have both this use case and this first initial set of use cases, they're returning relevant results. And uh it's more of a seesaw experience and every time you go through the seesaw, you're just covering more use cases. And this is a pretty huge difference between like a lot of teams when I might work with them, they might assume I'm going to label every use case that we have first and then we're going to tackle all the different corners.

Here we're sort of like tackling a use case, measuring we solved it, tackling a new use case and seesawing back and forth maybe for the life for years for the life of her product. I really like this slide a lot. I think it applies to all evals. I mean, even the ones beyond retrieval that we teach in this course. And I like the way that this mental model of seesaw, I'm going to borrow this. I like it so much with attribution. I really like it cuz people do get

confused. They're like, well, what happens when you change your evals?

Do you start all over again? A little bit, yeah. And so, this is a really good framing. Yeah. And one thing to add to that is you may need to go back to this earlier set of queries and like say, oh, these queries don't make sense anymore for this use case or the results are like outdated now. You may still need to keep these earlier stages alive. You're still trying to like seesaw forward as you're incrementally going forward.

Here we've decided with our search tool we're going to tackle a new use case. Now we're going to tackle initials for some reason and I came up with these fake queries. John Berryman, someone who works in AI, then Hamel and I know DT aka software Doug and these different use cases, right? So, I've added a new set of queries and I'm probably not adding three, I'm probably adding honestly like dozens when I'm doing this just to give you a ballpark. And this is what happens. This is the seesaw. So, these lighter blue lines are the things that already worked pretty well and now I'm adding examples of new use cases that I want to fix. When I worked at Open Source Connections and still to this day honestly as a consultant, I would help teams implement basically these tuning sprints on search where now the next sprint is going to be us working on improving these new queries while holding the old one the ones that already worked steady. So, this is the drop. We've added something to the set.

The overall average of all of these statistics these query statistics is worse even though the original queries are fine, we've just added new use cases we want to tackle. Yeah, I think another way even of saying this is your understanding of what good looks like changes. As you keep doing these, keep looking at data, the bar is raised. And so, when you raise the bar of like what good looks like, that arrow's going to go down cuz you just raised the bar. And you keep raising the bar.

>> way. So, yeah, I mean, there's different ways to think about it. Yeah, so like I said, I used to do sprints of this. And here is the we had the drop and now we're fixed. So, we're back up to a better mean. We've somehow fixed this use case and it's better. The other thing to think about with this to take it up a notch is everything I just did, you don't even necessarily need to think in terms of having labeled search results. There have been situations where like literally all I did with teams was, let's just try to measure whether or not search changed on the queries that you intended it to change.

So, if I just went through that whole cycle before and I had a set of queries I was mostly happy with like these light blue ones and then I introduced a new population of queries, I don't even need to have a relevance measure. I could literally just take the before and after of one of these queries and say, did it change? And that's kind of a head fake for people. That's kind of confusing. But you can see here if all I cared about was not hurting the queries that were existing, I literally could flag this change and be like, something changed in a query I should have kept still cuz this query was already making me pretty happy. So, one way to think about getting started with search, I literally would do this on search teams where yes, we would have like evals, but we would also would focus it on our highest priority things that just changed and do something different when we're evaluating. So, this is a bit of a safer change. This is a less safe change cuz I've in the safer change I've changed the queries I introduced or added. I maybe at least I'm probably impacting something that before was not

working well. For better or for worse, who knows? But at least I'm keeping the things stable that are already stable.

Already working well. This is another way to think about evaluation with search. And when you get these and I've seen this independent this process independently replicated two or three half a dozen times now where when you have this and you're like about to launch into whatever whether it's an AB test or you're going to like release update your tool into into your agent, literally taking these high priority queries that changed and doing side-by-side evaluation. So, instead of labeling search results, the whole dogma I just gave you like the entire dogma of search relevance. Now you should label search results as relevant or not.

Another way of looking at search which is equally and kind of measures slightly different thing and this is actually really can be really powerful for agents is to do side-by-side comparisons. So, if in all of that I noticed that the thing that changed was this here's a Doug Turnbull query. Um before after. I can now take this before and after. I used to call this a Pepsi Coke challenge because I often wanted it to be blind. It's very important to not tell people in your team that one is tested, one is control because they will choose want to choose the the new thing. And I would ask them, which one do you prefer?

The first thing is like, which of these search result listings do you prefer? And it's actually getting at more than relevance. It's getting at other potential things like search result diversity. So, search result diversity is basically you can return page of relevant results, but they're all identical. Like I searched for a job at a restaurant and they're all Chipotle jobs. Like, that's a poor experience even if everything's relevant. And then you could also look at things like are we returning useful information that tells the agent or the human that this is relevant. And that kind of shows up often in the traditional search context of things being actually seeing why in the search results something actually coming back relevant in the UI. And you can score all of these things in different dimensions potentially.

Relevance, diversity, these are kind of standard information retrieval metrics, but you can get to things that matter to your domain even. That tell you whether and why something changed and if you like that change. Kind of getting a gut check before you go and move to the next thing. There's a lot in regular search that's often implicit. Like, when I search for Doug Turnbull and I'm signed in, Google is assuming so many things. Like, Doug must want to see himself.

But, if someone else is searching for Doug Turnbull, there are these other implicit use cases that kind of go in there. And that's often why diversity is important. That's why often diversifying the search results in a for human searching is often really important cuz you're sort of modeling all the different possible intents when they haven't been explicit. I think we've uh beaten this to death. Don't turn this kind of thing into a up into the right dashboard. Make it an asset for the searching retrieval team. This has gone wrong when it has turned into that because suddenly it's not working as expected. Stakeholders get confused why we're changing things and adding We thought we were getting better, but it's seesawing. Like, that confuses people.

So, I think you really have to be careful with these kinds of graphs and letting them uh be taken as like ground

truth uh for the quality of the whole system. This is all pre-flight checks. This is all like are we feel good about these results before we go and out to production. I think the six big thing with in-flight with rag when we're out there, it's not an AB test. Like, usually uh in traditional search we might rely on an AB test. With in-flight evals in rag, we kind of have a different system. So, one thing is these search results in a traditional search, they just kind of invite engagement. And that engagement is really easy to measure, and we can make decisions for this emo mix query of like whether or not we these results are the ones that are before it or that like variant A versus variant B and its ranking. Rag, as I've said, doesn't really do that, and the information need is pretty complex. The other thing that's important and we just sort of talked about this with my earlier query, there's no objectively right emo mix.

There's uh many potential right emo mixes. So, ranking and diversity and all these things to try to encourage that engagement is really important in whatever AB test we're doing. When we're doing rag, usually we have like the users specifying everything about what they want for the most part. And if they don't, we kind of assume it's their own fault for not being specific. We're often in this more objective here is a prompt. We need a domain expert to kind of help us guide us to what the right answer should be. Would you say like in this uh more passive modality, you need to bring in recommendation system thinking? It could be. Yeah, so there is a even within search itself, there are these browse-oriented queries that are like literally like emo mix is probably a good example. But then within search you get longer natural language queries that start to look like prompts. And those start to feel less personalized and more like you really just have to like get the specific thing the user wants. So, it depends a lot on the query. And then yeah, all the way to the left of the spectrum is just recommendation systems. So, rec sys is kind of off to the side like we're not talking about it today, but it's sort of in the spectrum of how passive you are.

Like, are you just like on Amazon and might window shop something. So, this is especially true when you got these like highly personalized requests for actions. There's kind of an implicit assumption in here that the agent knows a lot about you and you've set this up. This is kind of a a level of maybe a a completely new level of AI evaluation you have to think about. But, we've basically set this up to harness this agent so that it really understands and does things as you expect. And I showed before that pyramid of autonomy. A lot of that autonomy, of course, is going to scale only in proportion. I mean, we see this with coding. We can do multi-agent crazy multi-agent systems or whatever, but only if we really have good guardrails. I shouldn't use the term guardrail, but like good feedback loops and things around the agent so that it does the right thing. In the case of search, this is also true. Like, search agents have a strong are strongly paralleled with coding agents. We are things like am I getting the right sunglass styles, pricing, brands, and all the requirements I need. Is that somehow in this validation step in the agent itself. So, I'm not thinking about AB test. I'm thinking about the stuff you're looking, learning in Hamoun Shayegan's class. Really thinking about domain expertise and like giving some good feedback on like whether or not the AI system is working as you expect it to. That's more the in- closer to the in-flight check that I think of with this kind of stuff. And the other thing, this is kind of like heretical in the search space, but I don't really obsess In addition to that, you kind of want the experience to be keep improving. I treat the business outcomes as a guardrail and not an objective. I treat the experience outcomes of the tool as more of an objective. And that's maybe like sometimes it's heretical to my clients and things, but the reason for that is if you can increase the pie that people like the share of the pie that people are excited to use this tool and it does a good enough job of doing whatever ultimate objective you want, whether

it's buying sunglasses or whatever, and you make the experience fun and great and it seems to be engaging, then if more users are going to come and the business outcomes are going to on not per session, but overall grow just by that increase in usage. But, all to say really it kind of gets back to what we were already talking about earlier, which is there are all kinds of reasons when you look at agentic trace that the agent either maybe the agent made a bad decision of what to purchase. And that's slightly outside the scope of just retrieval evals. Basically, with retrieval there's kind of two things that could happen.

Found what the user wanted, but at a high token cost. We were using grep as a search tool and it took 20 calls to find it. Or literally couldn't find anything even though the search seemed to be called correctly. Kind of to start to wrap this up, really just just gets back to feeding back into that earlier feedback loop that we already designed. Yes, that feedback loop is kind of a pre- I call it a pre-check tool. And really what it's doing is giving us as a team an increasingly more sophisticated intuition about what should come back in our search in these search tools.

But, you can see here in a recent trace there's some search for affordable sunglasses category gooder, and maybe for some reason that's not actually working. And uh we go through and we update maybe there's the promise that we remember earlier it was like maybe the tool is over not promising or documenting correctly what it cares about. But, more importantly, maybe we just need to add new evals and improve the search tool. So, this could point at a system where uh a situation where the agent's trying to use a search tool, but the search tool's misbehaving, and we need to look at that. And whoever's on our team need to sit there and be like, why are these not satisfactory to the agent? Maybe help the agent let the agent help us with that information. And then go back and see should we add this to our backlog of use cases we need to fix. And that just becomes another layer, another level set of queries that we iterate on in our sprint-based sort of system of going to tackle the next set of queries. Just to sum up, there's really a full picture here of there's an outer loop of making the entire agent AI system more effective. And then really what I'm talking about a lot is like translating that outer loop into an inner loop of iterating on the quality of the search tool itself in that seesaw pattern that we saw before. Doug is somewhat of a humble person.

I want to tell people a little bit more about Doug. Doug has a newsletter. You should totally subscribe to this newsletter because um I subscribe to it. It's just really good reminders and learnings all the time about retrieval. It's really good. You absolutely have to do it. It's just really good. Doug knows his stuff about retrieval, and I always learn something on every single time he does it. Doug does do a class on retrieval. It's probably the best class on retrieval. Like you said, it's the worst in the best classes. It's the worst because it's really very unique in the way he does it. Doug, you want to talk about what kind of things you teach in this if you don't mind me asking? I really am trying to keep up with the I see agentic search as the thing I mean, it's like where all the oxygen in the room of search and information retrieval is going in 2026. And you see people innovating in a lot of different spaces.

So, there's harness design. So, people are just learning how to build a good harness. It's similar to building a good coding harness, but building a good harness for search so that the agent continues to improve its results and gets better intuition about what should come back. But, then there's the tools itself. So, there's tools for like people are working um there's companies out there working on techniques that are really targeted at agentic queries.

So, there's that stuff. Um there's techniques called late interaction if you want something to Google. There are agents that are doing more code generation to basically improve their own ranking code. So, code generation like the constraints I discussed before about what the ideal sunglasses are, maybe that could be code that's generated that's like somehow the harness uses to evaluate the results that come back. Um up to to just throw other buzzwords around, there's recursive language models where literally the entire search is assuming I have a Python REPL. All the context is just a variable on a REPL. Or all of the the content I want to search is a variable on this Python REPL. And I'm generating tool calls to this REPL. It's often pretty bleeding edge the kinds of things we're looking into. Uh I try to make it as applicable to uh to teams as possible. But, I would say if especially if you're building rag or even if you're building traditional search and you kind of want to use agents to We also spend time with agents just optimizing traditional search. Like, if that's your interest and something you're curious about, definitely come and check it out.

That's really cool. If I'm not mistaken, you're writing a book right now? I wish I had the time to write a book. But, I do have books like AI-Powered Search and Relevant Search. AI-Powered Search came out 2024. That might be what you're thinking. Okay, yeah, this one right here. Yeah, exactly. That was a funny book to write because we wrote it before ChatGPT came out. So, its definition of AI evolved. >> Yeah, I have this book somewhere on this shelf. Oh, yeah, it's right here. Look, check it out. No, I'm not making this up.

Ah, awesome. That's awesome. It was not pre-planned. I had it right here. So. Oh, sweet. I'm flattered. I found that like the stuff you talk about in the book actually still applies. >> Oh, 100%. Yeah. Really useful. It's not like it decayed. Yeah, 100%. In fact, I teach a with Trey, the co-author, I teach another course on Weaviate AI-Powered Search. It's literally I don't want to say that's a bit less bleeding edge and more like how do we use machine learning to build ranking?

It's a bit more like the tried and true methods of search if that interests you, too. So, yeah, I try to we sort of have a class split between bleeding edge, we're not sure how all this stuff is going to work out, but some of it's probably going to apply, and more like tried and true methods that had seemed to we seem to keep coming back to with query understanding and that kind of thing. Makes sense. All right. Well, thanks a lot. That was really great presentation. I really liked it. That was really useful.

>> Yeah, definitely. And feel free to find me Software Doug everywhere, LinkedIn, Twitter, wherever you're hanging out. Sounds good. Thanks again.