EXPERT INTERVIEW

UPDATED
09/03/2024

Will Bryk, CEO of Exa, on building search for AI agents

TEAM

Jan-Erik Asplund
Co-Founder
jan@sacra.com

Published on **Sep 03rd, 2024**

Will Bryk, CEO of Exa, on building search for AI agents

By **Jan-Erik Asplund**

Background

Google, OpenAI, Microsoft and Perplexity are all competing to own AI search. We reached out to Will Bryk, co-founder & CEO at Exa (Series A from Lightspeed), to better understand the future of search without SERPs.

Key points from our conversation via Sacra AI:

- **The failure of ChatGPT's "Browse with Bing" add-on created an opening for companies like Perplexity ($20M ARR in April 2024), which drilled down into the prosumer search use case, finding contextually relevant search results, giving users a GPT 3.5-generated summary, and providing extensive citations.** "When ChatGPT used to have Bing browsing, it would start to use that, and I would always stop it. The reason: garbage in, garbage out… If the search results are not great, with a lot of SEO-driven spammy content, it'll actually make ChatGPT's output worse because it'll be conditioned on this low-quality knowledge."
- **The once-open web that Google first indexed in the late 90s is fragmenting into a patchwork of walled gardens, with AI**

companies forging data partnerships to guarantee their ability to scrape the most valuable media and news content, from Conde Nast (OpenAI) to TIME (Perplexity) to Axel Springer (Microsoft) to Reddit (Google). “TikTok might make it really hard to search over their content, or tweets could become harder to search because Elon is locking Twitter down. If that's true, then it's harder for a single organization to search over everything like we were used to in the past with Google. What might have to happen is either you have a bunch of different types of tools that each individually search over their categories, or organizations partner with those data sources. I think both will happen and are starting to happen.”

- **With every company launching AI apps based on proprietary data, AI-native search API companies like Exa (Lightspeed, $22M raised), Tavily (14K GitHub stars), Jina AI (Canaan Partners, $39M raised) and Perplexity have become a key component of the retrieval-augmented generation (RAG) stack alongside vector databases like Pinecone (A16Z, $138M raised), enabling companies to supplement their data with data from the public web.** “Our customers often bring their own LLM and prompt. We definitely think about those things too, but the vast amount of the value of our whole search system comes from the retrieval, not from the LLM."

Interview

Tell us about Exa—what was the key insight that led you to start the company and set out to redesign search for AI agents?

The core insight came three years ago. There was Google, and there was GPT-3. Google felt like it hadn't changed in a decade, because it really hadn't. GPT-3, however, felt like this new thing that could actually understand the subtleties of what you're saying. You could enter a paragraph, and it would understand exactly what you mean. Google, on the other hand, was using some basic algorithm with keywords and really wouldn't understand simple requests.

So the idea was: what if we could make a search engine that felt as smart as GPT-3? What if you could make a search engine that understood you, and no matter how complex your request, it gives you exactly what you asked for? It's a simple idea, but to this day, no one else has worked on it.

I think it's surprising to people that Google hasn't solved search. People say, "Oh yeah, Google has solved search," but it's just not true. There are so many different types of queries where Google completely fails, and I think those are often the most valuable queries.

What are some of those more complex queries you're thinking about?

Imagine you want to find people with similar ideas to you in San Francisco. You'd write an idea that you have and then say "people in San Francisco with similar ideas." Google is not going to give you a list of people. There might be 278 people who have had similar ideas online to you in San Francisco, but Google isn't going to give you that.

A perfect search engine - you type that in, and it immediately gives you 278 people. Not 200, not 300, but all the people that match. A perfect search engine would have high recall and high precision, so it gets everyone, and everyone it gets is correct.

We're searching for people, searching for companies. So, startups that are building futuristic hardware. Ideally, you could keep adding modifiers. For example, "that were started in the past 5 years." So, "startups building futuristic hardware that started in the past 5 years" - you should get a list of all the startups that match that. You could change 5 years to 4 years, and it perfectly adjusts.

Having a controllable search engine that just works, that gives you exactly what you ask for, is very novel. You could extend this type of thinking to not just people or companies, but products or research papers. Even just doing this over research papers would have a humongous impact on the sciences and all research, really.

Can you discuss the structural aspects that prevent Google from effectively handling those types of queries? Then, compare that to what's happening with Exa that makes those kinds of queries more manageable.

Google uses a keyword-based algorithm, whereas Exa uses an embeddings-based algorithm. I'm simplifying how Google works, as no one outside the company knows exactly how it operates. However, it clearly uses a keyword-based algorithm initially. It takes the trillion web pages on the internet and compares the

keywords in your query to those pages, finding the top matches. After identifying the top results, it may then rerank them using a more complicated method, possibly an LLM.

Of course, Google incorporates additional factors such as the quality and recency of web pages, as well as safety filters. It's more than just keywords, but it's fundamentally a keyword-based algorithm.

Exa, on the other hand, is an embedding space algorithm. We create an embedding from your query and compare it to embeddings of all the documents on the web. Comparing by embeddings is very different from keywords. Embeddings capture the meaning of the query and documents, so it's comparing by meaning rather than actual word matching.

In practice, this difference matters. For example, if you search for "startups working on futuristic hardware," you want a list of startups that are working on future hardware. Exa performs well with this query because it understands the meaning. Google, however, will see "startup," "futuristic," and "hardware" and find things containing those keywords. Often, these results are articles or listicles about new startups working on hardware in the future. Google isn't actually understanding the meaning, and it might give you a bunch of articles instead of what you asked for, which is a list of startups.

I don't know if this was recent, but I recently noticed "keyword" as an option in Exa. Are you using this kind of more keyword-based search to address queries for which maybe embeddings don't work as well as keywords? Or is it something else entirely?

That's exactly right. Embeddings are different; they work better on certain types of queries and worse on others. We want to work well on all queries, so we have a fallback to keyword searches. For example, "William Bryk LinkedIn" works well with keywords, and you don't need some fancy embedding algorithm for that. The ideal search engine would do both at the same time and be smart enough to know which one to use or perhaps a mixture of them. That's what we're working on right now. We're going to be shipping features like that very soon.

I'm curious about who you initially had product-market fit with. Who was really using Exa and incorporating it into their products? Can you tell me about that?

We started this journey about three years ago, in the summer of 2021. Our initial goal was to build an exceptionally good search engine, though we weren't entirely sure who the end user would be. We thought it might be consumers.

After going through Y Combinator and spending a year on research, we developed a novel search algorithm that was quite good. We released it on Twitter, and the first users were people seeking high-quality knowledge, for whom Google was consistently falling short. These users wanted excellent blog posts or essays about niche topics, and Google was performing poorly in that area. It was shocking for people to see that when they asked for blog posts about niche topics they'd always wanted to learn about, they received numerous previously unknown blog posts. It was really magical.

Our initial product-market fit was for consumers with more advanced search needs that Google wasn't satisfying. Then, ChatGPT came out about two weeks later. Interestingly, we started receiving requests for API access to our search engine. People wanted to integrate it into new applications they were building that ChatGPT enabled. At first, we responded that we didn't have an API, but as requests kept coming in, we realized something significant was happening. We discovered a whole world of AI applications that would need to search for themselves.

It turns out that AIs are very similar to those initial consumers who wanted high-quality information. AIs consistently seek high-quality information, unlike consumers who might search for things like "Taylor Swift boyfriend images." We evolved from serving consumers who want high-quality information to serving AI systems with the same need.

In terms of actual companies, initially it was startups building AI applications that needed to search the web, such as writing assistants or chatbots. Startups were the first to adopt this because the technology is so new. Now we're starting to see bigger companies, even those with 400 to 500 employees, building AI applications that need search capabilities.

You can think of it as waves of complexity in these applications. The initial challenge was simply getting an AI application to work by integrating an LLM. Now that many companies have achieved that, they want to move to the next stage: improving their LLM. One significant way to do this is by connecting it to the world's

knowledge, because an LLM that isn't connected to the world's knowledge is far less capable than one that is.

The first iteration of ChatGPT with web browsing access was somewhat disappointing in terms of the quality of information it retrieved. It seems to have improved a bit since, but overall, it still leaves something to be desired. What's the challenge for companies like OpenAI and Anthropic here? Do you see them as potential competitors or customers or both in the future?

Well, to your first point about browsing, it's surprising that came out a long time ago. I remember when ChatGPT used to have Bing browsing, it would start to do that, and I would always say, "No, stop." The reason: garbage in, garbage out.

If the search results are not great, with a lot of SEO-driven spammy content, it'll actually make ChatGPT's output worse because it'll be conditioned on this low-quality knowledge. It's really important to get high-quality information, which is why our customers often switch from using an LLM with Bing to an LLM with Exa. Exa provides much higher quality knowledge, and that really matters.

To your second point about these providers, we definitely see ourselves as search infrastructure that can power all sorts of applications. We want to partner with everybody and have people build applications on top of us. Building and maintaining a search engine, and doing research into it, is a very complex process. We're a research lab trying to make search a lot better, and there are a lot of resources that go into that. It's much easier to partner with a company that's doing it. Many of our customers want search capabilities for tens of billions of pages. They can develop algorithms to search over it themselves, or they could just use us as an API.

To go back to something you said about how Exa works: Is there a manual process for deciding which pages to turn into embeddings? How does that work? I'm assuming you're not turning every page that's out there into embeddings.

Right now, we are crawling a subset of the web. We try to crawl the highest quality subset that we can. It turns out that most of the web you really don't care about and probably don't even want to see. In our experience, it's better not to see it. Most of

what you care about is in, let's say, billions of pages, which is actually kind of cool.

Consider how many companies there are - maybe 50 million. How many people are online? About 1 to 2 billion. How many research papers exist? Around 200 million. When you add these things up, you don't actually get more than a few billion pages. A lot is dominated by people and news articles - there might be a few hundred million of those.

So, most of the web you care about is in that high-quality content. However, there is value in being comprehensive, so in the long run, of course, we want to be comprehensive. But as a startup going from zero to massive, the right approach is not to try and be comprehensive from the beginning. There have been other search startups in the past that have tried that, and I don't think it's a good strategy.

What is the criteria for quality? I had the impression it was something to do with the number of shares or something along those lines. Is that roughly correct?

There are various ways to measure if something is the type of content someone would share. If someone has shared it, it's usually good. However, we are building a search engine that should be able to avoid terrible documents even if they are dumped into it. It would be a bad situation if our search engine couldn't handle low-quality content because eventually, we won't be comprehensive.

I don't think it's such a hard problem to avoid bad content or have some sort of quality metric. For every search, you could add this quality metric to the results and then filter out the bad ones, or you could pre-process everything with a quality metric, which is probably much better.

No matter what method we use, our team prefers more end-to-end methods with less fine-tuning by humans and more learned algorithms. For example, the way we train our embedding model is completely self-supervised, meaning we're just using data from the web. This approach aligns with what the research community has learned from training large language models: it's often better to start the model with natural data because you avoid the biases of manually crafted datasets.

In the case of the quality assessment, it's better to have a principled way of training a quality model than manually determining what is good content. Manual labeling will likely miss a lot of important factors.

Exa has a web interface, but overall, the company comes across as API-first with the web interface as more of a demo of the API. Perplexity seems almost like the reverse—how do you think about that challenge of building both a prosumer product and API at the same time?

I wouldn't say it's technically challenging to add the LLM at the end, but it does change the focus of the company and the types of customers that would want it. When you start doing API and consumer work, you're spreading yourself thin, so I think it's very helpful for a company to be more focused.

We believe that the vast percentage of the value comes from really good retrieval. Our customers often bring their own LLM and prompt. We could, of course, very simply run an LLM with a prompt for the customer, and maybe we'll do that. But I wouldn't discount the value of simplicity. Even though it's very simple to run an LLM, it's annoying to do it in production, and it's nice to have something that works out of the box.

We definitely think about those things too, but the vast amount of the value of our whole search system comes from the retrieval, not from the LLM. So I don't think it's hard, but it's more of a product decision.

You recently updated your pricing, making Exa cheaper to use. How do you think about margins and building a sustainable business model at scale?

First, we want it to be really easy for builders to build without worrying about prohibitive costs. I think this pricing change makes it a lot better for developers and companies, which is good.

Regarding the cost on our side, there are many things we can do. With computers, you can always optimize something when the moment is right. I'm not worried about margins. There's always a trade-off between quality and cost, especially with search. We could expose that option, having higher quality search that costs more, and lower quality search that costs less.

Just exposing that as an option is a really different philosophy from Google, for example.

What's interesting with Google is that no matter how complex your search or who's using it, it always takes 400 milliseconds to return, and you have no control over the quality or how much compute is put into the query. If I ask "Who is Taylor Swift's boyfriend?" or search for the homepage of Walmart, those are very easy queries. It returns in 400 milliseconds.

If I ask for all the possible ideas on the Riemann hypothesis in the past 10 years, it also returns in 400 milliseconds. Isn't that weird? An ideal search engine would realize that those queries are vastly different in complexity and allocate compute resources differently, or at least expose that option to the user.

This generally falls under the philosophy of Exa, which is that we want to give users complete ability to control the search engine however they want. It should be fully customizable with all sorts of toggles to really customize it for you. This is a very different philosophy from Google or traditional search engines, which are saying, "Let's make it so simple you don't have to think about it. It's just a text box; we know what you want." No, I think for power users, we don't know what they want, and they should explicitly tell us what they want as much as possible. It's like explicit versus implicit search.

Web search was monopolized by Google in the early 2000s. Gradually, it became fragmented across platforms like YouTube (videos), TikTok (short form), Yelp (restaurants). Now, we're seeing it fragment across use cases with Perplexity (complex queries) and Exa (finding the best content). Will there be a Google of "search for AI agents" or do you see it fragmenting similarly, and how?

That's a good question. It's hard to predict the future, obviously, but I'll give it my best shot. I think there are different trends at work. One trend is that the internet could potentially become more closed off. For example, TikTok might make it really hard to search over their content, or tweets could become harder to search because Elon is locking Twitter down. If that's true, then it's harder for a single organization to search over everything like we were used to in the past with Google. What might have to happen is either you have a bunch of different types of tools that each individually search over their categories, or organizations

partner with those data sources. I think both will happen and are starting to happen.

Another trend, though, is that even if the internet starts to splinter off, there will always be some large percentage of the internet that is completely open. There is a huge amount of value in just perfectly searching over that open content. It's not really a worry to me; it's more like a sad thing. It would be nice if you had one place where you could search over everything and combine information from different places, completely opening up the flow so you could have perfect retrieval. But it's very likely that things will start to be cut off.

Yet another trend is that it's just more efficient for a single organization or a few organizations to do the search as opposed to lots of mini ones because you get benefits of scale. If you're a bigger company, you get to have more people to do more research to train smarter models, or you have more compute to train really smart models. There are just types of algorithms that only a larger organization would be able to discover. I want Exa to be that organization. There's also economies of scale, where it becomes cheaper for the organization. So, assuming that data is accessible, I expect there to be a couple of winners, not a huge number.

Google's search dominance made it the primary way that people discover new content online—and spawned the birth of an entire industry devoted to getting your content seen better in SERPs. How do you think about the shift in attribution and discoverability with AI agentic search vs. traditional search? How do content creators and publishers have to think differently about AI search?

I love thinking about the downstream effects of search engines on the internet, which are huge. It's possible that Google has had more impact on the world because of these downstream effects than because of the search itself. Google has incentivized people to optimize for a keyword search algorithm in order to be found, which causes people to write crappier articles or be more sensationalist. There are many weird downstream effects.

The more specific question you're asking is about the impact on content producers. There are definitely interesting things to consider here. Imagine we move from a world dominated by keyword search to one dominated by embedding search or

meaning-based search. Content producers will optimize for being found, so if they're being found based on meaning, they'll try to optimize the meaning of their content. This feels healthier than optimizing for keywords because they're actually trying to write articles that address the meaning people are searching for.

Perhaps a better way of thinking about it is that our search will be optimized for high-quality content. This will incentivize people to write higher quality articles. I could see a world where this type of search becomes extremely popular, and all the content on the web starts to become higher quality. That's such a beautiful potential result. I'm not being utopian; I think it's very legitimate. If the dominant search engine in the world only cares about high-quality content, content producers will make high-quality content because if they write poor content, it won't be found.

Regarding attribution, if the dominant search engine doesn't make money through ads but through per-query usage, there's an opportunity for content producers to share revenue with the search engine if their content was used by an application. I think that's a much healthier ecosystem because content producers are being rewarded for providing value as opposed to being rewarded for distracting consumers' eyes.

Another interesting downstream effect I've been thinking about is that you won't need news feeds in this world anymore. Theoretically, if you have perfect search, you won't need to scroll through a news feed because the search engine will give you exactly what you want. As a worker in AI, I currently need to go through Twitter to know what's going on in the industry, but there's a lot of unrelated content mixed in. With a perfect search engine, I could just get a synopsis of the relevant information and move on with my life. I love the idea of no news feeds, and there are all sorts of other downstream effects we could discuss.

What are the technical challenges that need to be solved for you guys to accomplish what you've been talking about? For example, scaling to the rest of the web and supporting millions of AI agents that are potentially using search?

I mean, the hardest part about what we're doing is not scaling. I actually think that AWS makes things so easy - it's kind of like just scaling up. A lot of these search systems are highly parallelizable, and we could totally scale up to everything. I don't

think that would be the right product decision right now, but we could.

The harder part is building perfect search, which is our goal as a company. All the technical challenges around that are significant. I don't want to minimize scaling - scaling is really hard, but it's a problem that's solvable. Whereas perfect search is a research problem, and it's not clear how to do it. We're on the frontier of AI for search and retrieval, in the same way that other AI research labs were on the frontier of generative AI.

Our hardest problems revolve around how to achieve perfect search. We have a lot of ideas, and it's very clear we could do a lot better than our current system. We're very excited to roll out those kinds of improvements.

There are different lenses through which to see how our search could get better. One is being able to pour more compute into the query. We now live in an amazing world where you can actually dump compute into systems and they get smarter. This wasn't true 20 years ago. If Google had said, "We're going to put 10,000 hours of GPUs into this query," they wouldn't know what to do. It didn't make sense because it was a deterministic algorithm - it always returned the same thing. Whereas now we have transformers that could be much bigger and run over lots of things. There are many ways to put lots of compute into a query, so I think there are clear wins there.

Another improvement is handling super complex queries with many modifiers. For example, "people in SF who have worked at search engines before, who have experience in Rust, who would probably be entry-level engineers." That's a complex query with many modifiers, and we're thinking a lot about how to handle that.

I think the right way of thinking about it is to take the analogy of SQL databases, where you break down a query and handle each part, then take the intersection. But this is more of a neural database, where some of the required components are fuzzy and require embeddings. We call it "neural SQL" internally. You can think of what we're doing as trying to make a new type of database that can handle arbitrarily complex queries and then put the web inside it.

I think treating our system as a database that you filter is the right approach, because Google doesn't do it this way. When

you type in "people in SF who've worked at search companies before," there might be a few hundred people who match that. But Google will say at the top, "30,000,000 results." There aren't 30 million people like that - there are only a million people in SF total, so definitely not 30 million who worked on search engines. Google is not treating the web like a database that you filter; it's treating it like a keyword matching problem. This is a totally different philosophy.

What people actually want, even if they don't realize it, is a database they can filter. We want comprehensive knowledge over any topic, and that's what we're aiming to provide.

Cool. If everything goes according to plan for you guys over the next 5 years, how do you see Exa looking? And how will the world be different?

Exa, at that point, is perfect search over all knowledge. I would say at that point we've gone beyond the web's knowledge. We'll have an Exa satellite, or many satellites, scanning the Earth and incorporating that into the search. That's a pretty big deal - literally perfect knowledge - and all the downstream effects of that.

I think science will be a lot more productive because of Exa. Any scientist can get exactly the previous experiments that are relevant to her experiment in the moment, in real time, and modify different words. It perfectly changes, and you have an LLM that's super smart, like GPT-7 or whatever, that's synthesizing all this information perfectly. You're getting real-time reports about the world's knowledge as you're doing your experiment.

This will have downstream effects on the internet at large. We talked about higher quality content on politics; I think people will be way more informed. A lot of problems in our society are actually just because people aren't aware of the possible solutions or good arguments. If those good arguments were presented to people, they would actually say, "Oh wait, that totally makes sense."

So yeah, it affects science, the internet, politics, your daily perceptions of the world, and your daily activity. You become way more productive. For example, it took me many hours to find an apartment. Theoretically, it should have taken me a few minutes because I would state all my preferences, and then it

would just process it. I mean, obviously, I might have to visit the apartments, but it'll be a lot more efficient. So everything just becomes more efficient.

Disclaimers