EPISODE 1091

[INTRODUCTION]

[00:00:00] JM: At a customer service center, thousands of hours of audio are generated. This audio holds a wealth of information that could be transcribed and analyzed, and with the additional data of the most successful customer service representatives, machine learning models could be trained to identify which speech patterns are associated with the successful customer service worker. By identifying these speaking patterns, a customer service center can continuously improve with the different representatives learning the speaking patterns that are successful. In the same is true for other speech-based tasks, such as sales calls.

Cresta is a company to build systems to ingest high volumes of speech data in order to discover features that correlate with high-performance human workers. Zayd Enam is a cofounder of Cresta and he joins the show to talk about the domain of speech data and what he and his team are building at Cresta.

If you would like to advertise on Software Engineering Daily, you can send me an email, jeff@softwareengineeringdaily.com. You can reach more than 30,000 engineers every day, and we'd love to have you as a sponsor. And if you are a listener who wants to become a paid subscriber to the show, you can go to softwaredaily.com and click subscribe. On Software Daily, you can also find information about different topics. You can find episodes that relate to one another, and we'd love to have you as subscriber. Thanks for listening.

[SPONSOR MESSAGE]

[00:01:35] JM: Scaling a SQL cluster has historically been a difficult task. CockroachDB makes scaling your relational database much easier. CockroachDB is a distributed SQL database that makes it simple to build resilient, scalable applications quickly. CockroachDB is Postgres compatible, giving the same familiar SQL interface that database developers have used for years.

But unlike older databases, scaling with CockroachDB is handled within the database itself so you don't need to manage shards from your client application. Because the data is distributed, you won't lose data if a machine or data center goes down. CockroachDB is resilient and adaptable to any environment. You can host it on-prem, you can run it in a hybrid cloud and you can even deploy it across multiple clouds.

Some of the world's largest banks and massive online retailers and popular gaming platforms and developers from companies of all sizes trust CockroachDB with their most critical data. Sign up for a free 30-day trial and get a free T-shirt at cockroachlabs.com/sedaily.

Thanks to Cockroach Labs for being a sponsor, and nice work with CockroachDB.

[INTERVIEW]

[00:02:58] JM: Zayd Enam, welcome back to the show.

[00:03:00] ZE: Thanks, Jeffrey. It's awesome to be back.

[00:03:02] JM: The last time that we spoke, you were doing research at Stanford. Tell me about your research and what you came away with from that research.

[00:03:13] ZE: Yeah, I was focused on how to rebuild machine learning to improve office productivity. How can we build tools that help people be more effective in the office? That was the predominant thing. We did a lot of user studied and build tools and software for different types of office work.

[00:03:32] JM: You eventually came upon this idea of Cresta, which is built to incorporate AI into educating customer service workers, call center workers, salespeople. Help me understand the problem with the call center or the contact center workflow that you identified.

[00:03:53] ZE: Yeah, absolutely. The problem there is a problem that we see, a problem that you see across many different workforces. Basically, the question is how do you make everyone as good as your best person? In any kind of sales team, any kind of sort of these kinds of

contact center environments, you have folks that are really good and other folks that are new and maybe are looking to sort of gain more experience. How do you take the expertise of the best people and help everyone perform at level the best person? That's one key challenge that's in the space. The second is folks are still spending tons of time, tons of time doing tedious repetitive things, doing things like filling out forms in Salesforce or other CRM's and doing order clicking and these kinds of things. How can you help automate and abstract away the repetitive and tedious bits of their work? Really, two major problems of this space, like how do you help people be fast at the really tedious bits of their work and how do you help them be good at the bits of the work that are really unique, that are really unique and creative to the type of work that they do.

[00:04:57] JM: That sounds like a really broad domain. What specific subset of that broad domain can you focus on and develop a product in?

[00:05:10] ZE: Yeah, absolutely. It's a very broad domain and it's a very sort of big idea. Really focused, hyper0focused on the use case that we have is we started with basically inbound sales chat. So you have these large sort of sales conversations that occur over chat and you have teams of 100 people to 200 people for large companies that are selling software, or telcos, or companies that sell retail products. The challenges you have on those teams is that you have some people that will take a conversation and convert it 3X to the level of somebody else. What are they doing in on conversation, that makes them so much better.

What we're able to do is go in and look at the conversations for the top performers of last year, collect the hundred thousand to million conversations that happened over the last year and identify which conversations lead to successful outcomes. You have a successful outcome and you're able to identify that this conversation led to that successful outcome and they're able to identify what are the behaviors that the person did on the conversation. In real-time, you're able to prompt people. Here's what the best person would've said at this point in the conversation. What that does is that gives the right thing to say at the right point in time and it really helps them have a better conversation with the customer and really sort of focus conversation, better conversation with the customer that leads to better revenue, better conversion, leads to ultimately a better conversation.

[00:06:33] JM: You start by identifying the people who are doing something right who are actually having success. So maybe you have some KPIs. Is that what you're saying?

[00:06:42] ZE: Yeah. We look at the sales outcome for each conversation. Did that conversation result in a sale or did not result in a sale? Then that becomes a training signal for us.

[00:06:52] JM: Once you have that training signal, then I guess you need to have this backlog or you need to have a bunch of data associated with what might've led to that outcome.

[00:07:05] ZE: Exactly. Basically, we start in the space of companies that do sales within these 20 minutes or 30-minute conversations where the whole sort of conversation is the transaction. Think about if you're sort of reaching out to a retail company and you're looking to buy a kitchen sink or you're looking to buy a new phone plan or you're looking to buy sort of accounting software, the sort of context of the conversation contains everything about what you're looking to buy within that conversation. Then we're able to prompt the salesperson and the support person on what's the best thing to say at each point in the conversation.

[00:07:42] JM: Now, that's sounds really tricky to construct to this flow where you can give people recommendations for what to say in the middle of a conversation, because you even if you have all the success cases of what led to a sale for the reps who are having a lot of success, the conversations could go in so many different directions and you could have maybe somebody who's having success because of the way that they inflect their voice or the way that they pause in the conversation. Can you find enough signal in the noise of just verbatim text that they're saying?

[00:08:28] ZE: Yeah, absolutely. Because what ends up happening is you have stuff which is this sort of –The sentiment of the conversation, the tonality, all these things that you're referring to. But if you look at the real meat of the conversation is the semantics of what you're saying. What are the questions that you're asking to truly understand the customer? Let's say you're – I'm coming in to sort of interact with you and trying to figure out I asked about a kitchen sink. The right response, sort of a really good salesperson will use that as opportunity to understand really what is the kitchen remodeling you're trying to do. What is the bigger project that you're trying to achieve and what is your goal with that and asking the right questions they'll

understand, like what are you trying to achieve? When they make a recommendation for a product, it's really understanding you as a person and what you're trying to achieve and really helping you get the right solution. That ends up being really understanding both the product and understanding how to really – Sort of the tacit knowledge of how to understand and get people to open up and could build rapport people on how to connect with them in a way that where that new salesperson might not be good at. So, that's a piece.

And so it turns out that, really, this part of that – That turns out to be a really big piece. You point out an interesting tension there, which is like, fundamentally, a lot of conversations, if you go to super – If you look at the different types of sales process, you have sales conversations that are super simple. Think about like one click Amazon purchases to like these really complex enterprise $10 million, or $100 million dollar deals. On that continuum of sales, there is definitely some sales that's super complex, super one-off, but there's also some parts of sales that's super simple and repetitive. Somewhere in that middle is a sweet spot for what we really focus on, which is when you have very large contact center environment where you're having these sales conversations and you're helping people perform them more effectively.

[00:10:26] JM: Okay. I'd like to talk about the data flow and the engineering side of things. Let's say the first problem that comes to mind is you've got, let's say, a hundred sales reps and you want to identify the top five sales reps. How do you build a system for collecting the information on those sales reps and then identifying who is closing the most deals?

[00:10:55] ZE: Yeah. This is often we're integrating into the existing underlying infrastructure, which is the contact center infrastructure and the CRM infrastructure, and we're able to basically get a mapping of conversations and the sort of outcomes recorded in the CRM of what was the outcome of that conversation. That becomes a mapping for us for the success or failure of a conversation.

[00:11:18] JM: So you can just integrate with the CRM integrate with Salesforce or whatever?

[00:11:22] ZE: Salesforce, and then you need to get the actual conversations and the transcripts from the actual contact center infrastructure.

[00:11:28] JM: Those are already typically set up. You already have access to those transcripts.

[00:11:32] ZE: Yeah. Those are integrations, and those are integrations that are built and that we need to – That we build. So they're not exactly simple to do, but they're I guess simple to think about, but to build reliable production-ready software obviously a different challenge.

[00:11:49] JM: Okay. Then the transcripts that you can pull in, you now have transcription and you have the reps that you've identified that correlate to those, or correspond to those transcripts. Tell me about how you look through the transcript identify what in the transcript is most relevant. What is leading to the sale?

[00:12:12] ZE: Yeah. What we've done is we've effectively trained a deep reinforcement learning architecture, which is looking to generate – So it has objective where it's looking to generate the next response and it's using the global loss of whether the conversation resulted in a sale or not resulted in a sale at the very end. So in some sense, if you look at some of this game playing AI, it is using a similar – It uses a similar approach where you have the success or failure of a particular game played and you're able to sort of predict the next best move that the game should make based on like sort of ultimately what would lead to a higher high rate of success or failure. It's a similar approach that we're using here to generate the next utterance, but then not only the next utterance, but what is the sequence of utterances that will result in a successful outcome.

[00:13:04] JM: And then how do you build the model of recommending chat messages and recommending what the other agents should be recommended to say?

[00:13:20] ZE: Yeah. Then it becomes an interesting challenge there as well where you're now predicting the response. What is the optimal response at that point in time that would ultimately lead to a successful outcome? But then you need to feed in as well what is the agent's sort of click through rate for a particular suggestion. For example, different people have different speaking styles. So you feed that into this large, very large scaled transformer model that's taking to context what is the current context of the conversation. What is this agent's ID? What is it effectively encodes? What is their speaking style and what will lead to a successful outcome for the conversation? And it's producing a response or a generated response that then sort of

being used as here's a response to use in the conversation for the next step of the conversation.

[00:14:11] JM: And can you tell me a little bit more about the engineering behind building the model that those reps are going to be using? Because I just want to understand, how do you create a reinforcement learning model that can adjust to all the different directions that some conversation could go in?

[00:14:37] ZE: Yeah. This is, again, sort of goes back into sort of – You're really finding out and scoping out the problem, the right problem that you're going after. If you approach certain domains where here's a huge degree in variance and the types of conversations happening, like open domain dialogue, that's almost impossible to solve. But if you really target a specific domain and – Specifically, in the domain that we have, for these kind of a conversations, there's a particular approach to the conversations that the best salespeople, the best support people use, that leads to a better outcome. Those are conversations where actually the reps don't let the conversation meander. So they're really focused on here's what the conversational flow is for the optimal thing, and they really bring that conversation back to that flow.

So, learning from that, learning from those behaviors and learning what the optimal flow for the conversation is, then lets the model basically say instead of having the conversation meander into one of a million different directions, let's really pull it back into what really needs to get done. And that's why it's a tractable problem for us to solve.

[00:15:37] JM: It's tractable, because in a typical sales flow, there is almost like a flowchart that the sales rep is trying to follow. And your goal is to find the language that you should be recommending to the sales reps at any given juncture in that flowchart.

[00:15:59] ZE: Exactly.

[00:16:01] JM: If we talk about the reinforcement learning, the system for building these improved models, can you describe the reward function, and the policy, and the state management system? The reinforcement learning model process, now that we've kind of outlined that you're trying to go for this flowchart model.

[00:16:28] ZE: Yeah, I think that's something in terms of the exact policy in these things would be covered under our IP. I'm happy to share what I can. So I can't unfortunately share too many details on that piece. I think the key underlying technical challenge there is there's a few challenges, like one challenge is how do you sort of segment? For any kind reinforcement learning to really work, you need to discretize the action space to a sort of a reasonable – Reinforcement learning that's really worked has worked in really discreet actions spaces. The challenge with NLP is how do you really discretize the action space in terms of what are the set of actions a person can take that will lead to success? That's a key challenge that we solve. Then how do you really – Once you do that, how do you really sort of generate responses and conversations to really be contextual to the conversation? Based on that discreet action space, how do you use that to generate the response for the conversation?

[00:17:26] JM: Could you just tell me a little bit more about the engineering behind the development of those models? Like what frameworks you're using and just the general process for pulling in the transcripts and pulling in the other information and making this all usable by the sales teams.

[00:17:47] ZE: Yeah. Absolutely. Basically, from a machine learning perspective, we use open source very much, right? Like TensorFlow and PyTorch. Those are really like best-of-breed sort of tools for us to build upon. The actual sort of data that we – Basically, the processes is we ingest the data, the historical data over the last year. We use that to train on models. Roughly, it takes us a few days to basically train the models. We collect the transcripts, collect the outcomes and train a model that's looking to generate the sort of optimal response at each point in time based on the success or failure of particular conversations.

So, it's ingestion process. We train the model. Then we deploy it to production. In production, then we're going to get real-time stream of the actual conversation. It's like we're pinging an API that's calling that server in real-time and predicting a response based on the context of the conversation, the agents that's making the call and sort of what is the optimal response. It's making a response in production. It returns as API call, and we feed that back into our interfaces and into our software, and it's providing a real-time prompt there.

[00:18:56] JM: What is happening under the hood when an agent is talking to a customer? The input from the customer needs to be translated into some kind of intent or some model or some set of more discrete parameters that can be responded to by the machine, whatever machine learning model you've trained.

[00:19:26] ZE: Yeah. Under the hood, what's happening is the model is identifying – It's identifying from the conversation what is the intent of the customer and it sort of creates these – It's extracting entities. Figuring out the intent of the customer and the intent of the conversation and it's using that to alongside the full, just on sort of structure, the actual text of the conversation to predict the response.

If you look at it, especially if you look at some of these really large scale transformer language models, things like in the last three or four years, these things have really started working, where even in 2016, 2015, it was still dubious at best whether these language models are actually working. But some of these recent advancements that have happened in the last two or three years have been so significant that your – I mean, you have coherent generation on the scale of a sentence or two sentences. Not so much at a paragraph, but definitely in the scale of a sentence or two sentences, you have coherent generation of text.

It sounds very unsatisfying and you say it's like, "Oh! You just feed it input text, and it gives you output text." It's like, "Oh! There should be some more engineering involved here. There should be like some kind of fancy ontology and like all these things. You need to have this and this and these components, but it ends up working. It ends up working. Because what effectively happens is that you really need large data. For us, what really matters is we need to pre-train on a very large corpus of data beforehand and then we need to really fine tune that model on that specific domain of the company that we're working in. But then it turns out that a simple algorithm just had a large scale of data ends up working really well without sort of too much complex tinkering under the hood to sort of map-out all these things.

[SPONSOR MESSAGE]

[00:21:12] JM: Airtable is a new way of creating software. It's a low-code platform with the ability to become code-heavy. And through years of development as a modern approach to a

spreadsheet, Airtable has now evolved to become a modern approach to a database backend. If you've ever thought of your database as a spreadsheet and wanted to treat it that way, that's how Airtable got started. Of course, Airtable is now much more than that. Air table has introduced blocks, which is a system for building rich application components that sit on top of the Airtable backend. And more recently, custom blocks, which is a system of JavaScript and React components that work through the Airtable SDK to enable developers to make their own functionality on top of Airtable. You can make a modern, authenticated, real-time cred app for users or admins or for yourself and you can make really anything that you would want to make out of a database, and it really saves you time if you're looking to get started in a low-code fashion. You can enter the Airtable custom blocks hackathon at airtable.devpost.com and you have a chance to win up to $100,000 in cash prizes. Get started by defining your database backend in Airtable and then move up the stack to building a fully-fledged custom block using the Airtable SDK. Win your share of the hundred thousand dollars cash prize at airtable.devpost.com. Try out a new way to build software with Airtable. Thank you to Airtable for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:22:56] JM: The models that you train, do you have some pre-trained models that you're just ingesting additional data into, or every time you integrate with a new customer, is it a brand-new greenfield system?

[00:23:14] ZE: Yeah. We have pre-trained models that we train more on open domain problems and sort of more fine-tuned conversational problems, and we sort of bring those models in when we're deploying to a new customer.

[00:23:28] JM: Okay. Do you have like different models depending on the kind of thing that's being sold? If I'm selling some healthcare widget to a hospital, that's going to be a different sales flow than if I'm selling – I don't know, derivatives. I guess derivatives are not exactly a product that you might be talking about sales for. But, I don't know. Twilio enterprise contracts, something like that?

[00:23:55] ZE: Yeah. Basically, our approach here is that we've built sort of – We've architected to be the same software across all customers, but the models for each customer are different, because the data that we train each customer on is a different set of data.

[00:24:12] JM: In reinforcement learning, like with Alpha Go or other models that play computer games, there's typically a discrete set of actions that the agent can take. In this case, you're dealing with natural language, because you're talking about people that are talking to each other. You got some customer or potential customer that's talking to an agent. There is an infinite number of actions that could be taken, because it's natural language. How do you define the pool of potential actions that could be taken?

[00:24:50] ZE: Yeah, absolutely. And that's a key challenge to solve in order to make it work. I think the key insight here is that, that actually, conversations –There's a lot of domains or the conversation is actually very repetitive across the conversation. For these kinds of closed domain problems, the conversations are very repetitive. So you have the same types of things being set over and over again. Because of that, you can really cluster types of responses and cluster messages and conversations into semantic intents. Through an unsupervised approach, you can effectively get an 80/20 rule where you can capture about 80% % of the intents through unsupervised clustering that really gives you and defines your action space.

[00:25:36] JM: So you can cluster – Basically, you can cluster any given input and, say, this piece of input is probably associated with this action. It probably corresponds that the person is trying to take.

[00:25:53] ZE: Yeah.

[00:25:54] JM: How much data does it take to train a model?

[00:25:59] ZE: Yeah. I think with some of the advancements that have happened with fine-tuning, that's also dramatically changed. Where before, like you needed to train things – Well, the common thing, which is training things from scratch. In the NLP, I think something like two years ago, it became possible to fine-tune. That changed the amount of data that you actually need.

For us now, like a sweet spot for us is something in the order of tens of thousands of conversations where that's enough variance and variability in the conversation and the outcomes of each conversation to be able to train a model to predict or predict optimal next steps.

[00:26:35] JM: Do you do any simulation to improve the models? Do any kind of like inflating of the data to improve the results?

[00:26:46] ZE: Yeah. Internally, we're basically taken the approach or – This is an engineering approach, and the way we think about the company is that we build tools to make it possible to onboard your customers in the matter of we want to get to the point where we can onboard a new customer in a matter of minutes. The way to do that is, firstly, recognizing that machine learning and just building machine learning, sort of these kinds of dialogue models and all these things, it's on a self-service approach. If you like start from day one and decide that you want to be a platform and you want to make it self-service for everybody to do it, you're not can actually be able to solve the problems for a customer because there's too much sort of nuance and knobs and all these things to change in fiddle with to really get things to work really well. But if you take the approach that you're going to go into a the customer, really demonstrate a huge result for them.

For our customers, like Intuit and Cox Communications, where we demonstrate hundreds of millions of dollars of incremental revenue, you can really go in and figure out how to get this stuff to really work for them and how to get the models to really shine and really improve sales performance. Then what you can do is you can build toolsets around that to really automate the process of onboarding a new customer to the point where you want to be able to get someone who isn't necessarily an engineer to be able to onboard a new customer by themselves, because they have the ability to do everything with –Do everything from simulate, sort of train a model with a one-click button. We have that sort of now where – We connect the APIs and these connectors for different chat platforms and voice platforms. Automatically infest the data. Sort of gets into the format that we needed. And with a one-click button, they can click it, start a training job, and then it generates output for multiple examples of test set conversations and it's like a

one-click dashboard for them to sort of see how it's doing. Being able to compare multiple models, sort of doing AB tests against multiple models.

Then for them to be able to go in and sort of mark certain things as outputs that aren't sort of ideal, and so that services training signals where it's like, "Okay, this is not the ideal output here. This is the ideal output," and there's some supervisory signals that you can get from that. It's basically building tools to help people more effectively train, sort of ingest, train models and test models, deploy them. That's our approach. If you can do that in a way where you can enable a non-engineer to basically be able to do this, you can really then scale out that approach. And that's really – That's how we're thinking here.

[00:29:13] JM: What's the biggest challenge in dealing with the high volumes of data? Specifically, the NLP side of things? Is there a lot of manual work, or like what are the workflows of figuring out and cleaning and labeling all the data?

[00:29:31] ZE: Yeah, there's a lot. There's everything from the data aspect, which there're key challenges there. Then everything to serving a production, like really, really big models in production at a high throughput and low latency, because you need to get under 100 millisecond responses. There's like key challenges there in terms of solving those sort of production challenges and those key challenges in terms of just training.

If you look at the data, there is all kinds of challenges there. Everything in terms of building a pipeline to training a model, you're training it on a very large dataset and you're often training these models on GPUs or TPUs. So it's like how can you build workflows that really streamline the process and really make it really efficient to train a model quickly and iterate on a model quickly? Because at the end of the day, if you look at the economics, the biggest cost is always the developer time. Obviously, the sort of person building the model, it's obviously their time that's going to take the most. That's going to be the cost. How can you sort of make it possible to train models across multiple TPUs in parallel to sort of reduce the time reduce that time and so that they can run multiple experiments and multiple iterations? How can you sort of ingest data in a format that makes it sort of basically easy-to-read and quick to sort of access while like you're doing random training? Basically, random batch training?

The advantage you have as text luckily is not – It's not a scale problem like Hadoop, because text is actually pretty information-dense. Where like with even like a very large scale text data, that will be less than hundred gigabytes, like very, very large. Like you download all of Wikipedia, and it's 10 gigabytes of data. It's not a Hadoop level problem. But what it is a problem is that you need to do a lot of competition against that text to really train a model that's effective, and that becomes a key challenge. That's becomes a key challenge in training, and then that becomes a key challenge in production where you're trying to now serve models with low-latency, high throughput without assertive compromising the quality of the output.

[00:31:30] JM: What are the frameworks that you're using?

[00:31:32] ZE: Yeah. We use predominantly TensorFlow for training. We use TensorFlow and PyTorch, but predominantly we use TensorFlow, because we find that that sort of framework is more robust for really high throughput, high throughput production environments, where Tensorflow is effective for us there. Then the rest of our stack is Postgres, Kubernetes, React, other parts of the stack, which are equally as challenging and interesting I think engineering problems, but definitely there is a focus these days on the machine learning side of things, but we think the engineering challenges across all stack are definitely really, really big.

[00:32:06] JM: Well, tell me more. Where are the most acute engineering challenges that you're seeing today?

[00:32:11] ZE: Yeah. I think, broadly, there is a few things that are – There's a few key challenges I think if you look broadly in the space. One challenge is like it's effectively a challenge. It doesn't sound sexy, but it's really key challenge. I think it's a fundamental problem for everything. There're fundamentally different data platforms across like all of software, right? Everyone has different schemas for the way their APIs, the way you integrate into their APIs and like how this sort of record a transcript and what are their sort of ways of – What are the integration APIs in terms of how is the transcript recorded? How are these things done? How do you get a new conversation and how do you close a conversational? All these things. It's a fundamental challenge to be able to build a general architecture that can integrate into the whole sort of gamut of different platforms that exist in this space. And it requires sort of understanding all the sort of edge cases that appear and like all the different ways that these

things work can occur. And it's actually really hard to build something that is really generalizable. You have to sort of do almost like a 5-80 rule, where you like build you build five [inaudible 00:33:20] five platforms, you can get about 80% of the value. That becomes like a key challenge of being able to build that effectively.

Then the second thing is like once you do build that, it's like at the end of the day, what the product is the product is the product is in front of a user. The user interface really matters. How do you sort of prompt someone and how do you give them interactions in a way that really is non-obtrusive to the workflow and really helps them sort of have really effective conversations. So, so many things matter there. Everything from latency, to the way you prompt them, to like how easy it is to like turn it, like sort of say no or how easy it is to use it? What is friction, and all these things? Those are like key challenges in building really great interfaces.

I think JavaScript, which is very interesting language, and like we use Typescript, but there's a lot of key challenges there in terms of making it perform, making it something where it's really usable by the salesperson or the support person, but that also sort of really – It's like an architecture they can build upon and then you can continue to iterate, because you're constantly experimenting with new sort of ways to prompt users. Constantly sort of building an architecture that you can sort of build features and sort of try new things and do different things within the system. So that becomes like the – Your sort of investing in something and then you build a bunch of tech debt, because you're trying out a new feature to see if it works. Once it works, then how do you sort of incorporate that and bring that into the fold? That's a key enduring challenge for us as well, because we're constantly experimenting.

[00:34:50] JM: Is it an issue to get low latency inference on a given message that you get from – A salesperson gets a message from a customer, and there can be a lot of text in a message. Is the latency of an inference ever an issue?

[00:35:11] ZE: Yeah, it is. So you have to be careful about particularly long conversations and how long it would take to run inference on them and can you do approximation for the very long conversations, at the very long context? Because if you feed token by token of like a thousand, like a hundred message conversation into a lot of transform model, the latency might be up to a minute. So what you need to do is approximate certain parts of the conversation and really only

use the inference for the more sort of recent parts of the conversation. So that becomes a key issue.

Then, secondly, like sort of doing it in production, you obviously have a cost constraint in terms of being able to deliver the product. So you need to be able to deliver it in a way where you're not just horizontally scaling out GPUs, where you have a more efficient way to batch them, batch the responses that are coming in from multiple agents. So that becomes another key challenge where you have 100 agents using it and they're also sort of sending requests. How do you batch them in a way that can then be sent to a GPU and get a response back in a really efficient way? Because GPU's and TPUs are not single – They're sort of optimal for batches of batch workloads where you group a bunch of requests together and serve them all at once, as supposed to sort of doing it one-off at a time. That becomes a key challenge for us.

[00:36:23] JM: If we talk more about the engineering stack, what are the cloud provider services you're using and do you have any interesting anecdotes about building Cresta on top of cloud infrastructure?

[00:36:41] ZE: Yeah. It's interesting. My cofounder, Tim, who is – I thought I was a good engineer, but then I met Tim, and I realized I'm not a good engineer. At some point in – I think everyone has some point where they meet somebody much better than them and they're like, "Okay. This is no longer –" Like yourself impression of yourself is out the window.

I knew Tim from the PhD at Stanford and we had sort of interacted a bunch throughout the PhD. And then I started working on Cresta, and one day I messaged him like, "Hey, let's grab dinner. Just have some updates I've been working on the company, some updates." So we grabbed dinner at this Thai restaurant at Redwood City and like showed him the demo of what I was working on, all these things. And like I had small little office at the time. So like after dinner, he came back and we were just playing around and hacking around and he like loved it so much, he joined, he started the next day. He showed up the next morning and like we've been working together ever since full-time. But the next he showed up and he said lake what are the AWS – I was running everything on EC2 instances on AWS, and he's like, "This is – What are you doing here?"

Tim had just come from OpenAI, and I think that day he's decided, “Okay, I’m going to set up Kubernetes.” So like he spent the next hour, two hours just setting up Kubernetes and just getting everything up and running on Kubernetes. It was, one, the speed of it of getting us sort of re-architected entirely on Kubernetes was insane, because I think he had seen the scale at OpenAI and sort of seen how sort of certain things can be done at scale when you have the right architecture in place, and he knew how to do it. He knew how to do it effectively. So he was able to do that really effectively.

But then he became something that went to the market, we actually found a lot of companies that didn’t want to business with Amazon. So, for example, a lot of companies that compete with Amazon actually don’t want you to use AWS for your cloud provider. So that really saved us in a few of these sales deals, where the customer is like nonnegotiable. You guys can’t use AWS. We’re like – What have probably killed those deals for us. But because we had architected on Kubernetes, we could then quickly spin up new clusters in GCP and Azure and very quickly became multi-cloud.

Yeah, and so there’s like a few things like that that really sort of the right decision early on really helped the business really grow. And so that was interesting. Interesting piece. I think it was a good decision at his part to do that early on.

[00:38:55] JM: What are – Specifically, when you’re architected on Kubernetes, this might be a naïve sounding question. But if you are architected on Kubernetes instead of sing raw EC2 instances, what is beneficial about that?

[00:39:08] ZE: Yeah. So there are a few things. One, it gives you sort of agnostic – It makes easier to be agnostic to different clouds. I mean, obviously, your database is still not packed in the Kubernetes container. So there are still things that you need to figure out for each cloud instance that you’re sort of bringing up to speed and all these things. So you have to figure out database and all these ingress and all these things. But it makes it easier to be agnostic to different clouds. If you have particular requirements or like, say, a particular cloud provider is too expensive for a particular workload, it’s much easier to switch between different providers.

Then I think just is became really simple. Another key point where it really helped us was that it became – I think at some point, I remember one day early on in the first six months of the company, we signed our first major deal with a customer and they had a particular quarter target to hit, because sort of they had particular growth estimates that they've given to Wall Street. Internally, they're red alarms to the company, because we're not going to hit this number. We need to do everything we can to sort of really hit the number. That's why they brought us in with such urgency. Like usually companies don't do business with one person, like just early stage startups, but they needed the sales number. So like they needed to onboard like a couple hundred agents like in a matter of a day or something, and it as like the first sort of spike of traffic that we would see. Our solution was just to simply horizontally scale out our service, because the sort of models are independent of each other, because each model is self-contained. So in production, you can just scale out the – You can scale out the inference models, and we're able to like, overnight, scale up to like a thousand X traffic that we had never seen before, because it turned out to be very expensive, because horizontal scaling is not the most efficient solution. But that was another thing where it allowed us to quickly scale up to that peak, and that was a key advantage because Kubernetes was containerized already and it became really easy to sort of autoscale the pods.

[00:40:57] JM: Wow! Have you noticed anything interesting about how workloads for each company, each customer that you work with. How do the workloads vary across those different companies?

[00:41:12] ZE: Yeah, it's interesting, because, fundamentally, it's a pulse of US business, right? Because the conversations fundamentally represent transactions that people are doing with companies. For certain companies, they have peaks during Christmas seasons. For other companies, they have peaks during tax season. For other companies, they have peaks during like sort of summer when everyone is moving. You have like just different workloads at different periods of time depending on like when people are sort of consuming that product in the economy. You can imagine, at Christmas season, all the retail companies have just huge amounts of traffic that they just – That's where they make huge amounts of money during Christmas season. Then summer is when a lot of people move. A lot of companies that sort of service homes with like cable plans and all these things and internet plans, they have a lot of traffic, because a lot of people are moving in that time period. Tax season, all the accounting

companies and all these folks have a lot of traffic at the begging of the fiscal year. That's like when people are buying those kinds of products and it becomes a reflection of underlying conversations and underlying transactions happening across the United States.

[SPONSOR MESSAGE]

[00:42:18] JM: There are two ways to add analytics to your application, you can build them yourself with basic charts and dashboards using free open source charting libraries, or you can use a comprehensive analytics platform from a partner that you trust. If you've tried to build it yourself, you know that free actually is not so free. There are hidden costs like time, and maintenance, and technical debt, and those hidden costs can really add up.

Check out Logi Analytics. Logi Analytics is developer grade embedded analytics solutions and they make it easy to create branded dashboards and report the scale within your own application. You can stop wasting time piecing together analytics and allow yourself to focus on your core application. You can go to logianalytics.com/sedaily and you can get a demo to see what is possible with Logi today. Go to logianalytics.com/sedaily. That L-O-G-I-analytics.com/sedaily.

[INTERVIEW CONTINUED]

[00:43:32] JM: Do you have any perspective on what is going on right now in the economy? I don't know how many customers you have and what sense you have a pulse of the world of business. But I mean, in the post-COVID world, you have kind of a divergence of opinions on what's going on with the economy. Some people think the stock market continuous to go up because the fed is printing money and just buying lots of assets and it's kind of an artificial thing or it just kind of propped-up. But there's also just maybe a case that the digital economy has become unmoored from the world of restaurants and construction sites and such and such. Maybe we have an economy that's actually resilient to this kind of thing. Do you have any perspective on what is going on with the economy?

[00:44:24] ZE: That's an interesting question. I mean, I think, broadly, I think there's a lot going on. It's hard for one particular perspective to state – The thing with the economy is that – Or with

any economy is that it's just such a complex system that if you're trying to simplify it, you're not doing the thing justice, because it's such a complicated system. I won't be able to fully address what's going on with the economy, but there are interesting trends I'm seeing in the sort of market based on the data that we have, which is interesting, and I use that to sort of guide my thinking on it. But again, the economy is a very complex system. So it's hard to say exactly what's happening.

One trend we did see across many customers, which is that all of a sudden, overnight, all these retail customers had to shut down their in-store operations and they had to then move effective – They have these revenue forecasts for in-store sales and then to move entirely to digital channels. All of a sudden, they have to sort of drive sales with their digital channels that would come through in-store. We actually had customers who bought us in this period that, basically, we needed to sort of convert their in-store salespeople to online and phone sales people. It's like how do you train them, onboard them to have great conversations over messaging and phone? Basically, we had one company [inaudible 00:45:38] actually that was – They sold mattresses and they were sort of – They announced their Q1 earnings last quarter, their Q1 earnings and their stock went up 30% even though they shut down all their stores. They had actually brought us in and we had demonstrated 24% increase in revenue per conversation. Yeah, it was like a $7.3 million incremental revenue for their business within a two-week period. They brought us to really do that to really drive their online and phone sales. They exceeded the analysts' expectations and their stock went up by 30% the day they announced their earnings.

That was something where it was like a dramatic sort of improvement for their business where they had to shift. So like their sales are still down relative to like – Because their whole in-store operations are shut down, but their sales are still down overall. But they're doing much better than expected, because they're bringing in the right technologies to really help augment their workforce. That's one thing that's happening.

Then the other companies are actually benefitting from it, where you have all these like telcos that are actually sort of seeing huge, like unprecedented volume, where because now everyone's at home, every needs better internet plans and better home entertainment and all these things. So they're upgrading their internet plans and upgrading all these sort of parts of their packages. So they're seeing some really great sort of traffic and volume.

You have parts of the economy, definitely SMBs are the most hit. SMBs, especially like sort of SMBs that are in-person locations, and that's fundamentally the part of the economy that's sort of definitely the most hit. So it's a function of basically their loss of foot traffic and they lost basically the folks sort of transacting with them. That part of the economy is sort of right now somewhat kept afloat by the PPP and all these sort of government programs, but it's not clear that sort of when the economy does come back, that it's going to be like it rebounds back to what it was before. It will likely be some kind of sort of delayed U period where sort of it takes us a while to get back to the state that it was before.

[00:47:42] JM: Amazing. Can you tell me more about the problem space of the company? Now that we've talked through a lot of the minutia, just zoom out and tell me what has been really hard to solve. What has been the areas of the company that maybe are giving you pause or sort of starting to identify like the prototypical issues that you're going to be grappling with for the foreseeable future?

[00:48:14] ZE: Yeah. To give you some context, last time we chatted, I was doing my PhD and I dropped out of the PhD. So this is a very focused problem to really increase sales performance, but for me, sort of the major thing that's happening is that there's a major shift in the way people are going to work and it sort of happens every – It happens every hundred years or so. If you go back to early 1800s, sort of a single person could reap a quarter acre a week per day in like 1802 and 1803. Then by 1820, a single person can reap a hundred acres a week per day. It was this 400X improvement in productivity because of the horse-drawn reaper. The technology that came in was a horse-drawn reaper and all of a sudden unlocked this massive productivity boon and it made it possible for farms to be built that could support full civilizations and cities could be supported and farmers can now support many more people and just change the way modern civilization operated.

Then you zoom forward 100 years ago to early 1900s, and that's when the first manufacturing was really happening, when you had sort of cars being built. But the early cars in 1902, 1903, were all sort of cars that were bespoke, really expensive and really unreliable. They were built sort of for rich people, for like individually – Not many people could afford them. They weren't a great business to be in. And it wasn't until Ford and General Motors really built their first

scalable manufacturing principles with the sort of assembly line and they made it possible to build a car, where before in 1902, it took 800 hours to build a car. By 1920, you could build a car in 2 hours and 30 minutes. All of a sudden, it made cars affordable by the middle class, and everybody, many more people could afford a car and it changed the way modern US society can be functioned, because now you could build roads. People go to restaurants. The way that we sort of transported and commuted and how we spent our recreational time completely changed. The whole nature of cities changed.

The same thing really is happening in office work. If you're looking at office right now, the clerical work that people did in the 40s and 50s and then started using software from the 80s and 90s, that's like stuff that doesn't necessarily have to be the way it was a hundred years ago. If you go forward 50 years from now, will people still be doing the same repetitive tedious things like in terms of filling out forms and like doing those repetitive things? Will people really be using the same type of software to do their type of work? It was like, "No." This is not going to be the case. This total type of work is going to change and machine learning is going to help sort of the nature of this work change.

For me, to drop out, it was like, "Fuck!" I want to be a part of this. I want to be able to say that I sort of helped changed the way people worked and I sort of helped, sort of was a part of the shift to the way that people did this kind of work. With machine learning, you're seeing it now. In the last 10 years, if you look at the US economic productivity in the last 10 years, it's actually been flat. If you measure economic productivity by the total GDP output divided by the total number of hours worked by the US economy. It's a very simple number. But if you look at it for the last 10 years, it's been entirely flat. We actually have not seen any GDP improvement.

There all kinds of hypothesis in like as to why that's the case. We've had so much progress since that time, but it just happened. We haven't seen the numbers. But like, for this, this was like, "Here. You go into a company, increased your sales performance by 24%." You see the stock price go up by 30%. I mean, it's like, "Wow! You actually had impact on GDP and you actually made – You sort of built the economy. You're sort of helping improve the economy and help improve the GDP by sort of building tools and helping people be more productive. So you're sort of seeing that now and sort of the same shift is going to happen. You see these 24, 30% gains, but you're going to see 100X gain in the way people work and the productivity of

people in 10 years in terms of how office work is thought of, where you might have sort of person doing 100 times as much effectiveness of work that they were doing now. That's really sort of really exciting thing to me, and that's why I get so excited about what we're working on and why I dropped out.

[00:52:05] JM: Wow! Okay. That's really cool. I mean, your bright future for the world of machine learning actually feeling present in every element of our lives. I can't wait for that to come. You do feel it in small ways, but I guess this will be a boiling frog, right?

[00:52:26] ZE: Yeah. That's interesting analogy. Yeah. I think, yeah. It will be as pervasive as software is right now. All the folks in the 90s, like Gates and all these folks are talking about how microcontroller would be in every single device that you have. There'd be computer in every device that you have. People sort of put off as wishy-washy, like for some computers are like what we're going to have. First, it was mainframes and personal computers. In the 90s, everyone started saying everything will have a computer. That actually turned out to be true, because you have microcontrollers in pretty much every single device that you have like, everything from your headphone to your cars like all have microcontrollers and computers in them. And it's just something that it just happened. Everything became an electronic device.

I think the same way – It's not like everything is going to be machine learning device, but you're going to see it happen in ways where like you wouldn't have even imagined that experience before. But machine learning is going to unlock it. Before, if you asked somebody in 1899, would you like a car, or like – They'd be like, "No. What the fuck is a car?" The roads are even like paved. How am I going to drive a car to get to the next town? Because roads are like – They're not even paved. It's a two-day journey.

But then once the car sort of came, then that sort of created this experience where now all of a sudden it made sense to invest in road infrastructure. It made sense to invest in highways and all these things. And all of a sudden, it sort of changed the way people experience things and lived. I think the same thing will happen to machine learning, where machine learning will unlock new experiences and make it possible to do new things. That's going to unlock entirely new things that we wouldn't have even imagined to be possible. One day it's going to happen like,

"Oh! Shoot. Everything is a machine – Machine learning is sort of impacting so many experiences in my life or making it possible," that we'll sort of recognize that.

[00:54:03] JM: Okay. Just to wrap up, what are other areas would you be working on if you were not working on Cresta?

[00:54:12] ZE: Yeah. I have a list. I think there's a lot of really interesting things happening. I think one thing that's happening that's interesting. If you look broadly, I think healthcare is a very interesting market, but it takes someone who has really a stomach of steel to like go and decide that they want to spend the next 20 years in the healthcare market. But broadly, I think in that market, you have basically mass orders of inefficiency in that market where it's like it's just the way the whole system is set up is sort of highly, highly inefficient. I think that there are a lot of things that you can do there. I think, for example, one thing, you look at like – I think that same time as starting Cresto, is evaluating, sort of working on a sort of a digital pathology. Basically, the FDA has recently approved devices to do digital pathology. So started looking at sort of skin cells of these things, and it became like the first FDA approved device that digitally I've been working for. It wasn't digital at all. I even actually tracked that market, but something like that really unlocks the market opportunity, where all of a sudden instead of having pathologist sort of distributed across the world or like inefficient way where you have a pathologist at a local doctor's office. When it's digital, you can be much more efficient about it. You can built centralized systems or you can build a marketplace. You can sort of build these kinds of really efficient ways to be able to diagnose these slides. That unlocks major efficiency.

I think if you sculpt out hardware problems, like tele operation, I think there's a massive opportunity in tele operation where you can tele operate robotics and tele operate sort of these kinds of like kiosks and these things that are one-step further from the current automated experiences but are not so complicated that they wouldn't be feasible. I think that's really big market opportunity.

I think the challenge with some of these opportunities is that it's like business is all about figuring out what is the first thing you can do that gets you to like some kind of repeatable system and then how do you build up on that? That's like with Cresta, why we chose like a particular sort of sales and bouncing all these conversations as like the starting point, because

it's a great place to build a business. Then it lets you sort of build towards something greater. But for these kinds of things, you really have to figure out what is the piece of it that's really going to build a great business that will then let's you build up on it to sort of reach the full vision of the company. That becomes a challenge I think for any of these opportunities.

[00:56:30] JM: Okay. Zayd, thanks for coming back on the show. It's been great talking.

[00:56:34] ZE: Yeah, absolutely. Thank you, Jeffrey.

[END OF INTERVIEW]

[00:56:44] JM: When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

Go to softwareengineeringdaily.com/g2i to learn more about what G2i has to offer. We've also done several shows with the people who run G2i, Gabe Greenberg, and the rest of his team. These are engineers who know about the React ecosystem, about the mobile ecosystem, about GraphQL, React Native. They know their stuff and they run a great organization.

In my personal experience, G2i has linked me up with experienced engineers that can fit my budget, and the G2i staff are friendly and easy to work with. They know how product development works. They can help you find the perfect engineer for your stack, and you can go to softwareengineeringdaily.com/g2i to learn more about G2i.

Thank you to G2i for being a great supporter of Software Engineering Daily both as listeners and also as people who have contributed code that have helped me out in my projects. So if you want to get some additional help for your engineering projects, go to softwareengineeringdaily.com/g2i.

[END]

