EPISODE 63

[INTRODUCTION]

[0:00:10.6] SC: Hello and welcome to another episode of TWiML Talk, the podcast where I interview interesting people doing interesting things in machine learning and artificial intelligence. I'm your host, Sam Charrington.

Let me start this show by sending a huge thank you to everyone out there listening. We've dropped a ton of great interviews over the past few weeks, but through your dedication, we continue to see a growing outpouring of feedback, comments and shares with each release.

If you're a regular listener, but don't normally send in feedback, we would absolutely love to hear from you. Please head on over to Apple Podcast or wherever you listen and leave us a review. Of course, a five-star review is appreciated, but what's most important is that your voice is heard. It lets us know what you like or what you feel we could improve on, and it lets those looking for a new machine learning and AI podcast know that they should join the TWiML community.

Speaking of community, the details of our next TWiML online meet-up have been posted. On Tuesday, November 14th at 3 PM Pacific time, we'll be joined by Kevin T, who will be presenting his paper; Active Preference Learning for Personalized Portfolio Construction.

If you're already registered for the meet-up, you should've received an invitation with all the details. If you've yet to join the meet-up, head on over to twimlai.com/meetup to do so. We hope to see you there.

Now as some of you may know, we spent a few days last week in New York City, hosted by our great friends at NYU Future Labs. About six months ago, we covered their Inaugural AI summit, which is an event they hosted to showcase the startups and the first batch of their AI NexusLab Program, as well as the impressive AI talent in the New York City ecosystem.

Well, we were more than excited when we found out they would be having a second summit so soon. This time, we had the pleasure of interviewing the four startups of the second AI NexusLab batch; Mr. Cleverest, Bite.ai, SecondMind and Bowtie Labs.

We also interviewed a bunch of the speakers from the event and we'll be sharing those discussions over the upcoming weeks. In this first episode of the series, you'll hear from Mt. Cleverest, a startup started by childhood friends James Villarrubia and Bernie Prat.

Mt. Cleverest is an online service for teachers and students that can take any text via the web and generate a quiz along with answers based on the content supply. To do this, they employ a pretty sophisticated natural language understanding pipeline, which we discuss in this interview.

We also touch on the challenges they face in generating correct question-answers, how they fine-tune their machine learning models to improve those answers over time and more.

Now, on to the show.

[INTERVIEW]

[0:03:15.2] SM: All right, everyone. I am here at NYU Future Labs meeting with some of the AI NexusLab companies. The first company in to the interrogation room is Mt. Cleverest. I'm here with CEO James Villarrubia and the COO, Bernie Prat. Welcome to This Week in Machine Learning and AI.

[0:03:36.2] JV: Thanks.

[0:03:38.3] SM: Yeah. Why don't we get started with an introduction to the two of you, your backgrounds, as well as the company and what the company is up to.

[0:03:46.9] JV: Sure. My background is really is an engineering statistician. I got a engineering degree from UVA and then Masters in public policy mostly focused on tech policy. What I found is the coolest most interesting problems out there to solve; weren't just engineering ones, they were political ones. They're big picture policy stuff.

I jumped into the Pentagon as a statistician and then briefly went to the White House, and then finally at DOJ, Then I got tired of DC and got the startup bug and came up to here to New York and helped run a cyber-security company for a couple years. Until eventually, this idea that we had about Mt. Cleverest finally piqued the right person's interest and they said, "Go do that." We said, "Great. We've been trying to get it going for a long time." We're really excited.

[0:04:31.9] SM: Thanks James. Bernie, how about you?

[0:04:33.9] BP: Sure. I moved up to New York right after college and we did some work in a startup in finance, where we were pulling data out of press releases to programmatic traders. This allowed us, or me to get into machine learning and natural language processing early on in my career, not as an engineer, but as a manager of engineers doing product work primarily.

From there, I bounced around to a few different places, again in product positions but very data-heavy, API-heavy. James and I, we've actually been friends for about 25 years. We met in kindergarten.

[0:05:04.6] SM: Wow.

[0:05:05.5] JV: Yeah.

[0:05:07.2] SM: It's the first time I've heard that.

[0:05:08.4] BP: Photo evidence too.

[0:05:10.0] SM: It used to be part of our pitch.

[0:05:11.9] BP: Nice. We've been friends for a very long time and had always wanted to work together. After doing so, we went to high school together as well and did some good work in a number of different areas, but this is the –

[0:05:21.6] SM: Transfer High School?

[0:05:22.7] BP: Yeah, we were in ROTC together, we were – James was doing the band in the newspaper and I was a member of that, and we were just all over the place.

[0:05:31.9] SM: Once in band camp?

[0:05:36.4] JV: Bernie never went, but we don't talk about that.

[0:05:38.6] BP: Yes.

[0:05:39.5] JV: No. No.

[0:05:42.1] BP: Yeah. This is an opportunity for us to work together.

[0:05:43.9] JV: Yeah, finally. Yeah.

[0:05:45.7] SM: Awesome, awesome. Mt. Cleverest. I love the name. It's actually rather clever.

[0:05:50.8] JV: Yeah, you know who did that.

[0:05:52.5] SM: What does the company do?

[0:05:54.7] JV: We actually started the company based off an idea we had seven years ago. My parents are both teachers. I actually still teach at UVA part-time with cyber-security. What we realized is that growing up, I see my parents deal with bad textbooks, textbooks that are handed down to them from administration that they had to bend to meet the needs of the classroom.

We were iterating through ideas and that was the one that I said, "No, if we could figure out a way to solve this, this would be really valuable," because my parents always were complaining about it. We didn't really quite know the approach yet and we didn't have necessarily the NLP and machine learning skills to really tackle it yet.

But as the industry has grown and our skills have grown we said, "Okay, this is something we can apply NLP, machine learning, neural nets and solve this selection problem." What Mt. Cleverest does is we can take any content on the web, URL, piece of text, then generate quiz questions on the fly on that text in a matter of seconds.

Then the real adventure though is not just creating the questions, but after that we track every interaction with that quiz, what seems to get wrong, what to get right, how long they take with each question, what questions and answers teachers like and don't like, what they want to use, feed that into a neural net, then improves the system and improves the quiz of the next person. It's actually, really at the end of the day it ends up being one big online textbook that actually improves itself, the more you use it.

[0:07:16.0] SM: Interesting. Interesting. I'm familiar with – there's a ton of research happening around question answering, which is you have a body of text, you start with some questions and then use AI machine learning to try to answer those questions.

You've been boarded that and you're using AI to generate the questions, as well as the answers and track – track and adjust over time. Is that a established research area, the questions asking piece?

[0:07:47.2] JV: Not so much. I mean, a few years ago, there were very few papers. At least now, there are some – there is some interesting work being done, but again nothing is substantial, because a lot of it has been driven by the question answering Alexa, Siri approach to machine learning.

The question that has – or what has been tackled in machine learning is this idea of taking a mass amount of text and figuring out a good question come around it, kind of the walk-in model jeopardy. To be able to parse mass amounts of text and generate one question, maybe a very well-formed question, but it's one question.

Our problem in this and really the heart of our tech is that we take a very limited amount of text and be able to generate 120 questions off of one piece of text, each asking about something different. Then the extra level of difficulty is then, "Great, you've got a question, you've got an answer that you've drawn out. How do I generate good wrong answers? How do I find the question – or sorry, an answer that is a good distractor from the correct answer?"

I put a multiple choice question in front of a kid. If it's so close to the right answer that it's confusing, that's bad. If it's so far away that they would never guess it anyway, that's also bad. You got to find that Goldilocks middle ground on good distractors, and that was really probably the heart of the NLP problem.

[0:09:04.2] SM: I'm imagining like a dial, where you can tune the question for difficulty? Or, I guess a big part of that neural piece that you described is about normalizing the question-answer set so that your average ends up where you want it to be, is that a good way to think about it?

[0:09:20.6] JV: Yeah. We can actually look at the readability, the text or industry standard, that scoring mechanism and take a chunk of text and figure out what grade levels this would be appropriate for. The origin text, we can actually guess at what good, what these questions are going to come out referring to in terms of like, is this a 5th grade text, or this is 7th grade text?

We're actually working on some tech to be able to convert something that is maybe written on a 4th grade level down to a 9th grade level, and then use questions that are more appropriate for that. That's in a coming release.

Yeah, it's being able to correctly pick questions that are appropriate for a grade level and answers that are appropriate for a grade level is a whole layer of complexity that a lot of other tech solutions don't really think about what is unique to the education space.

[0:10:06.7] SM: I think of educators as folks want to have a lot of control over their source materials, like how does this approach land for them relative to what they're used to?

[0:10:20.2] BP: We're taking this problem in two different ways. The first is initial product allows a teacher to bring in whatever content they would like. If they found a news article that is exciting and relates to their class, or if they have a page that they've been using and sending students to for years, instead of having that all based on paper and in terms of the grading, we're actually pulling that in and tracking that information.

On the other side, there is a search component that you've brought into it and we're working towards that. That is the broader vision of having a collection, a universe of content and then being able to rank and sort that. There is the create problem, then there is the ranking and sorting. We split those intentionally.

The control issue, it's something we want to get into. We know that it's a political battle in the lightest of terms, because it's almost not true politics. But yes, that is something that we see often. We're coming up with ways to either encourage behavior in a certain way to let go of that including automatic randomization of questions, or adding in questions from a different source that applicable to a source that was provided, so we're actually swapping in new information and using that as part of the quiz for students. It's not a standardization, but it's instead a way for us to help determine the quality of content, as well as the students understand it. There is a couple different problems there.

[0:11:41.1] JV: There's also a distinction in education between formative assessment and summative assessment. This idea that teachers get really – have a close hold on exams, because they want to be able to assign really clear grades and hold those students accountable, and they should be with parents that disagree with those grades.

They have to have a clear chain. But what has been lost is this idea of this formative assessment that I want to have homework that a kid can try over and over again until they get it right. Because actually been learning it, not just grading on their first guess.

What Mt. Cleverest is really focused is on its kind of try to get the teacher to not think about the quiz that we are providing as summative. As yeah, you're going to get grades and this is going to account to the report card, but more of a get good questions in front of a kid to make sure that they understand the content, and they can take that quiz as many times as possible, until they really grasp it or master it.

[0:12:33.2] SM: Okay. Can you talk a little bit about the pipeline that you used to deliver this from a machine learning perspective?

[0:12:40.8] JV: Yeah. It's funny, because we get a lot of questions about like, "Oh, can't this just be done by a machine learning model off the shelf?" Yes and no in a sense that we are using some kind of – some basic industry tried and true NLP models and neural nets.

[0:13:00.8] SM: What are some of those?

[0:13:02.3] JV: We're leveraging a lot of NAR for a name that's in recognition using some interesting math around word vectors, or I call it like the donut model saying, "Okay, I want to find similar words." But then I go up and they remove the words that are really similar to find those – again the Goldilocks distractors.

Those are the approaches that we're doing in the neural nets. We're trying lots of different things, because that part of the product is still – not nascent, but we're getting there. We're refining that. But the pipeline itself actually, what's interesting about Mt. Cleverest is that it's not just one huge big monolithic model, is that we have to build an NLP that can generate dozens of types of questions, so each one has its own pipeline; feeds that into a unified data structure.

Then we have to normalize questions against questions, answers against answers, formats against format. Even within the ranking system within one lesson, within one subject we might have a whole series of models. Each, again not necessarily a huge robust of crazy big models, but each doing a unique specific thing.

[0:14:01.4] SM: Can you elaborate on that? What do you mean by normalizing questions against questions? Are we talking about in terms of their difficulty and things like that?

[0:14:09.0] JV: Right. In terms of their difficulty. Imagine you're a teacher and you've created a lesson and you want to – we've generated let's say a 100 quiz questions. Now some of those are not going to be great, because the NLP isn't perfect. We don't know many products that are perfect.

We want that human in a loop feedback. We are capturing the up-votes and down-votes, similar to Reddit of what teachers like and don't like, but at the question and answer level. We get that, just that base level of how many people have up-voted, how many people have liked or disliked this particular piece.

But then we've got this student performance data on the back-end. It's like, okay when I showed this question in this format in this context, like this is how well the student did, given how long they took. Normalizing that data means that I have to take every interaction with every question or format or answer and then trying to figure out, "Okay, even though this question has been showed to 10 different students in 10 different ways, how do I judge its effectiveness? Is this a good measure of learning on this particular topic?" That's the normalization that we're talking about.

[0:15:07.4] SM: Is that some big batch job that runs every X day, or week, or something? Or is it something that you just trigger periodically, or –

[0:15:17.8] JV: We've actually made a goal of the product to not do things in batch, which was it bold going. It slowed us down a little bit, but we actually wanted – we set the goal from the user experience almost that we wanted the quiz to improve on a per interaction basis.

[0:15:34.1] SM: Every time a kid answers a question –

[0:15:35.8] JV: Exactly.

[0:15:37.7] SM: - it re jiggers the entire –

[0:15:39.3] JV: Yes.

[0:15:40.5] SM: Does that mean –

[0:15:40.7] JV: By the time the second student sees the quiz, it's actually improved and changed and learned from the results of the first student.

[0:15:47.8] SM: Is that learning, does that mean that you are retraining models and that whole pipeline, or does it mean that – are there some set of heuristics that you're using to massage weights, or things like that, without having to retrain all your models?

[0:16:01.5] JV: A bit of both. We don't retrain the model every run through that we do have some trail trained models worked in. But most of it is capturing the heuristics of the performance with those questions in those formats, and capturing that and then feeding that back in as additional information into the model, as we then randomize and select what we want to do and what we want to show to the second student.

It's the heuristics of the first student fed back into the same model tail trained a couple pieces, and then what you get of like, "Oh, tried this version of this content, or this version of this question with these answers in this format," instead of the old one. Because that old version, it wasn't a great test. But this one might be. It's a little bit a randomization, a little bit of design and experiments constantly trying to say, "Okay, reduce how many experiments or variations do we need to run before we figure out what the best stuff is?"

[0:16:47.7] SM: Okay. In terms of identifying the target content in the first place, is that – are the educators feeding URLs into the system to direct you, or are you doing some ML-driven crawling or something like that to figure out the interesting content of their –

[0:17:04.9] BP: At the moment, we're relying on the what we consider expert networks of teachers. We want the educators to be the one providing content that they have used in the past, and then we'll be doing that ranking in order to help them either bubble up what is better, or take advantage of the stuff that they're tried and through sources.

A lot of the open educational resources movement from the Obama administration has been very helpful, and that a lot of institutions, as well as a lot of individuals are now providing content openly available on their creative comments, license or IT license, whatever it might be, is now available to us and now behind the payroll and not under the umbrella of one of the large publishers or anything like that. It's taking advantage of that increase in trend.

[0:17:46.1] SM: Okay. Interesting. What has been the biggest challenges in pulling this altogether?

[0:17:52.1] JV: I think the biggest challenge was designing that data structure. Again, we have all of these different models all looking at essentially the same data and how you store that and how you store that in a way that can be kind of light-weight, pulled in the moment in between quiz questions.

That was something that I think both of us had to take a step back and say, "Okay, how do we store this data at scale, but as we're small, but also at scale building towards this bigger architecture?" Being able to track what I like to call the context mapping, but it was this version of the question with these answers showing like, maybe they're 10 wrong answers, but we only showed four and it was a true, false, or multiple choice with a none of the above not this.

Being able to capture all of that context and store that and then correct tease it out and then build a model against it, figuring out what that included and how to put all the pieces and data together, that was essentially – I think we have a model in the tool called a Quinstance. It's like quiz instance.

We got to generate a quiz, and it might have teacher preferences at one given moment. But then within that quiz, within that context we need to have an individualized version for each student that they experienced at that moment in time.

It's like, how do we refer to that? It's a quiz, but that's also a quiz. Quinstance became our term and it's been great. But we have a lot of those little things that we have to figure out on a way that like, "Oh, we have to structure this and store this in a unique way that I think gives us an advantage in the market that I think other people have really thought about data for education this way yet."

[0:19:25.6] SM: For folks that are – how did you approach that problem? Did you just stumble upon the answer, or did you just try out everything that was out there and see what worked? What did you end up doing?

[0:19:36.5] BP: These are sore subjects. We're running in circles here. It's okay, but he look at this.

[0:19:41.0] JV: Flashbacks. When we first started trying to build this, I was just like a wedding web developer. I actually built the whole tool without a lot of the neural nets. I hadn't quite gotten there yet. But I built my own parsing engine and NLP engine out of PHP, because it was the only language I knew. I had a lot of lessons learned.

[0:20:01.7] BP: From the –

[0:20:02.3] JV: Right. It was a terrible, terrible idea with – I think the front-end was like a Drupal 6. Right. Right, exactly.

[0:20:10.6] BP: I'm proud of these moments. They were learning experiences.

[0:20:14.9] JV: Exactly. I got 10 times better as a developer, and it jumpstarted I think a lot of my career just being having to suffer through that. Having a really good problem to chew on for a long time. Then eventually I said, "Well, no. Python is better for this. There is Java libraries that I can start incorporate. I need to stand up micro-services architecture."

Right now, we have I think 14 different micro-services, each running even different parts of the models, all talking to each other, learning this data set, caching some parts here and there. But I had to learn that all on my own starting with that poor, poor Drupal set. Yeah, it's trial and error.

[0:20:46.4] BP: Could've been Wordpress.

[0:20:49.8] JV: It almost was. I thought it was advanced when I found Laravel. I was like, "Oh, man. It's not Drupal."

[0:20:58.3] SM: In terms of the data store, did you like some document-oriented MongoDB type of thing, or like Cassandra, or what direction did you go for?

[0:21:09.4] JV: Because we didn't really know what we were doing and I think that's actually a great use case for Mongo, is that it's a turnkey solution and you could drop in whatever you want, and then there were a lot of lightweight or realms on top of it. You can change the model and it doesn't break everything. You just had to go back and claim some stuff later.

We've been doing Mongo for probably the last year as we moved to a more production level product. But we've now hit the wall and we're saying, "Okay, the model type have settled, we know what our structure is, we know what it needs to be, we've thought it out, we've been in the wild and see what happened."

Now we're actually moving to a combination of things. Probably some kind of Mongo or other NOSQL database for a lot of the document structure. Then probably Cassandra for a lot of the – like the interactions – the minimum interactions, because it's built this scale linearly, where everyone else stops at a certain point.

I'm really excited to move to Cassandra, but Cassandra also has some issues around search. You got to figure out how to get searched on. There's a lot of other different ways looking at postgress for other certain features, index and search there. We're actually moving to a more complex model, but again, each database has to fit a different part of the system and we now know how that system really needs to be set up to run on light speed.

[0:22:25.4] SM: Given that you're doing the interactive updating and all that kind of stuff, I'm assuming that's not so PHP. Is that like Spark, or –

[0:22:34.0] JV: It is almost, like there is a little bit of Python, on the backend some Java libraries that I haven't written, but leveraging some open source stuff. Primarily, the entire stack is written in javascript, leveraging Amazon Lambda and then react as a frontend.

We have made this as a if – the heavens opened up and the God said, “Yes, please. Go use this product,” and we had massive scale that we are set up to – I see them handle it. Scale breaks everything, but we knew that there is going to be so much compute that we wanted to make sure we could turn every little level. All those micro-services are most of them are speaking Lambda to each other.

Also means that the models are lightweight enough in some respects that we can offload some of that model computation to the browser. We have this javascript written neural nets, so I can actually load some of the early computation onto the computer for the teacher or the user of the phone, and then do the computation and just back the results and offload a good 20% to 30% of our load back on to the user. We’re not doing it now, but we are set up to do that eventually.

[0:23:34.0] SM: Wow. Are you using any particular open source library to do the frontend inference? Or did you write that yourself?

[0:23:43.1] JV: No. That’s a combination of Synaptech. I think Synaptech Dutch AS was the one we’re using right now. There have been many variations. I think first one we used was called Brain. There’s a lot of javascript libraries that are coming out where they said, “I want to write javascript and I want to do neural nets.”

They can do some pretty cool things on the browser, not necessarily the big scale stuff that we want to do on some of our calculations, but a lot of the smaller like within a document within a document, known document space; we can do that in a browser side and we got to shift to that model later.

[0:24:13.8] SM: Interesting. Walk me through – I’m trying to think through the way I’ve envisioned your process. You get these documents, you’re doing a bunch of what I think of as backend processing to come up with questions and answers and normalize and all that kind of stuff. What would you want to do on the frontend that would require running the inference smoothly? Or would take advantage of that if not required?

[0:24:36.1] JV: The first thing you do as a user is you drop in a URL, eventually to be a search term or URL But right, we’re focused on new content capture. Drop in a URL, then immediately we go scrape that content from that URL and then start parsing it. A lot of parsing work, right now we're handling the backend, but it can be handled by the browser. That is actually a big portion of the compute, because it's a lot of parsing.

But if that can be offload instead of 10 people doing it all on one of our machines or a series of Lambda functions, by pushing to that browser – because that's kind of – it's not proprietary stuff, it's not crazy complicated, but it just needs to get done. Someone needs to pull out, name identity, someone needs to break, do sense vary detection. All that stuff can be handled and we just want the results. Then we get to the nitty-gritty of the data on the backend.

[0:25:22.0] SM: It strikes me that if a lot of folks aren't doing that now, that's going to be a popular way to do cooperative compute. It's almost like having your users mine Bitcoin for you before legitimately.

[0:25:37.7] JV: The only other industry that I know is really doing that is Bitcoin and not like, "Oh, my advertising window is mining Bitcoin for someone in the Ukraine." When I read that I was like, "That's a really cool architecture." We've already been thinking this is like, "Oh," someone is proofing that this is doable, so it was a check in the box of "Yes, this is maybe where architecture should go."

[0:26:02.5] SM: Interesting. It strikes me that that could be a startup in it of itself. But solidifying its architecture, because there's all kinds of problems that you could run into of like the process is stealing all the compute and usability issues and stuff like that.

[0:26:16.0] JV: Yeah, if we figure that out, then we can shift to – that will be our second business once we sell this one.

[0:26:21.8] SM: Nice. Given all that technology, there's tons of folks, particularly here in New York City doing ad tech. What makes you different?

[0:26:32.5] JV: We've been doing this project for a long time, so we've been watching the ad tech market when the booms and busts – we're on the third boom right now that we've been paying attention to. We keep seeing investors getting burned, mostly because they keep investing this – what seems like a really cool new thing, but in reality it falls into two standard business models.

One, we call a warehouse model. Just collect as much information as possible, and then they get searchable. But if you look at some of the products out there, you find that those – the search algorithms are poor at best. They're not searching on real performance, just keywords.

You end up getting a big warehouse and you'll search for the word 1812, for example. You get 342 results. I'm not speaking about any product particularly. All of them are ranked 3.9 or 4 stars out of 4 stars. As a teacher, like I try to use that. I was like, "This is useless for me." Great. Am I going to open 342 PDFs and then read them and say, "Oh, I think this one would work."

Even if I learn, it's like, "Oh, this one actually was effective," I have no way to really transfer that knowledge back in and share that with my community. That was like one thing, okay we can solve that part.

The other business model is we call the wall garden model, which is more like the 1920s newspaper business model. Whereas, ad tech companies are essentially taking investor money, having authors to write content and then putting it behind a pay wall, just like a newspaper, traditional media.

I just like, "No, no. You can't read it. But I promise you education with that." Right. I was going to say it. You have this model where the incentives are misaligned, or it's that it's not in their interest to share what pieces of content they like and don't like, because you get like, "Oh, you buy this whole sweet of stuff."

Yeah, they want to improve it, but they're not necessarily helping the teacher do that and really enlisting the teacher's help, being honest about what is and is not working. If I ask someone at The New York Times, if I think their company, or the newspaper is better than a Washington Post, I know who they're going to say, I know what that answer is.

Getting real data and performance out of this traditional model is just hard. You got this high-quality hard to produce, expensive to produce new content model wall garden, and you got warehouses; high quantity and very low quality. The more stuff you get, the worse the user experience gets.

We said, "Can we tie quality and quantity together? Can we make it a positive feedback loop?" That's really where the AI and machine learning cuts in is that we are capturing at scale enough information to keep floating the best stuff up to the top, but searching through it and capturing it in a way that's not just traditional search. That means we can cut through the top and get quantity and quality at scale.

[0:29:12.7] SM: Have you found that a better way to rank the way you present content is by bubbling up what you seen with the interactions, as opposed to asking for an explicit star rating or things like that? Is that the direction you're going with this?

[0:29:30.1] JV: What I would say is that we know that star rankings are not working. It's been tried. We are capturing that data. But instead of just having someone vote on like, "Oh," like I like this piece of content. Instead, we can capture more interesting data about like, I've got this piece of content and there is a hundred questions in it and I've seen tons of people who have strong opinions about the top 20.

The ranking and sorting algorithm is a bit more off of the Reddit idea of how do you have these layer-nested pieces of information, how you float all of that in some kind of recursive loop of like, "Oh, now I know that this piece of content is better than this one and I'm not just using the up-votes and down-votes on those top level pieces, but everything inside." That's really what's enabled us to take the next level in terms of ranking. You combine that with performance data, and now you've got something that no one else is going for.

[0:30:22.4] SM: Well that's a very cool story. What's next?

[0:30:26.1] BP: If you take what we're doing right now and you really tease out that there is teachers using this system, what we can do is we can then take the data that the teachers and students have produced and bring that to institutions themselves and say, "Here is a view of your school that you've never seen before. One that would benefit you in terms of looking at how your students are actually learning, as opposed to just the output grades."

We're looking at, are the students digging deeper into the content? Are they actually mastering something as opposed to did they just memorize it and they're gone? Looking at the next phase is using aggregated data of students and teachers at the school level, at the district level, the local government level and that type of thing.

[0:31:12.1] JV: If you remember, we're talking about earlier about the being able to become normalize and standardized questions against questions, answers against answers, well there's nothing preventing us from using very similar models to rank students against students. Not just students against their classmates or against themselves, but against every other student who's ever touched the system.

What I think is really unique about the way that we're doing this is that even if students see different pieces of content and take different quizzes, we can compare them and say, "Okay, which student actually learned more as doing better?"

If you look at the way we are standardizing student for results for today, kind of standardized test, it's shove a student in a room on a Saturday and you put 300 questions in front of them and pray. You hope you get good data.

With Mt. Cleverest though, instead of that 300 questions, you can get 30,000 data points per student, per year, over the course of a year and you can finally get to this as Bernie alluded to the second order metrics of success that are the holy grails of education policy, of questions like did the student become more curious, about what did they become more curious? Did they learn how to learn over time?

For now, kind of where and what was – what are they most successful at. That's the stuff that a lot of this typical standardized test models can't get to. We know that at scale, that's where we think a lot of these data will be valuable, is that we can actually provide real standardized testing data to schools without them having to actually do any standardized testing. It's just part of the problem.

[0:32:37.9] SM: Awesome. James and Bernie, it was great getting to learn a little bit about Mt. Cleverest and explore how you pulled it all together. Fascinating story. I really appreciate you taking the time.

[0:32:48.2] JV: Thank you.

[0:32:48.4] BP: Thank you.

[END OF INTERVIEW]

[0:32:53.8] SC: All right everyone, that's our show for today. Thanks so much for listening and for your continued feedback and support. For more information on Bernie, James, Mt. Cleverest, or any of the topics covered in this episode, head on over to twimlai.com/talk/63. To follow along with the NYU Future Labs AI Summit Series, which will be piping to your favorite podcatcher all week, visit twimlai/nexuslabs2.

Of course, you can send along feedback or questions via Twitter to @twimlai, or @samcharrington, or leave a comment right on the show notes page.

Thanks again to NYU Future Labs for their sponsorship of the show and the series. Once again, thank you once again for listening and catch you next time.

[END]