Transcript

11 November 2025, 11:57pm

Si Chen 0:23

2.

All right. Hi, everyone, and thanks for joining. We'll be starting in a couple of minutes. We're just waiting for a few more people to trick in, trickle in. In the meantime, just feel free to drop a quick hello in the chat. You know, let us know maybe where you're joining from. I'm in Sydney. Daniel is in Singapore. Tell us a bit about your role and maybe your.

Focus or interest areas in AI and data space. So just as we give them a couple more minutes.

All right. It's got some people coming in now. Um.

Performance. So my name is C, very pleased to be your host today. I am the VP of Strategy at Appen where we're focused on helping our customers basically improve model performance and development through very high quality training data, evaluation data and we work closely with both Frontier Research labs as well as enterprise.

That are deploying AI in a variety of applications and industries. So the topic today is I think a really interesting one at the intersection of AI and data and I think it's a very challenging problem. So I'm sure a lot of people who are joining the webinar have seen examples of this, but we essentially see a lot of.

Improvements when IT comes to cutting edge models, very high performance when IT comes to public benchmarks. But this often doesn't reflect the reality when we start to put this into production. So essentially, how do we actually address and bridge that gap? So I'm very excited to introduce our speaker today, Daniel. Daniel is the Chief Data Scientist at the.

SAP Business AI unit. His team focuses on developing deep learning models for document processing and benchmarking AI models. Daniel's been at SAP for over 13 years. He's also an adjunct assistant professor at the Singapore University of Technology and Design and teaches at Heidelberg University as well.

We actually met last in person in ACL in Vienna, but I'm currently in Sydney, Australia at the moment and Daniel's in Singapore. So we have a very global crowd that's here today. But I was very excited to invite Daniel because it's very interesting to have that

bridge between research as well as real world applications.
And that wealth of knowledge, I think is is going to be very exciting for everyone who's attending today. So today Daniel is going to focus on sharing some limitations with benchmarks, techniques that can be used to better measure models when IT comes to true reasoning or usefulness in real world applications. And we'll talk about some use cases like information extraction.
And red teaming enterprise agents in terms of the format. So Daniel will present for about 35 to 40 minutes and then we'll open the floor for QA. So in this webinar there is a button for QA, so you can pop your questions in there as IT progresses.
Anytime during the presentation, but we'll address them at the end just to ensure that we've got enough time to get through Daniel's content. So that's enough for me in terms of an introduction and please join me in welcoming, welcoming Daniel.

Dahlmeier, Daniel 8:32
Well, thanks for having me. Thank you. Hello, everyone. Nice to meet you. Yes, I'm Daniel. I'm the chief data scientist at SAPAI team. I'm based in our wonderful labs here in Singapore. My background is in natural language processing. That's a field that has been.
Transformed a lot by AI. So when I go to ACL now, it looks very different from those days when I was a PhD student and present my research there. But yeah, very happy to be here to talk a little bit about what we do with regards to benchmarks, how we look at model evals.
Within SAP, how we approach this topic and also some of the research that we have been doing in that field and hopefully that is interesting or helpful to you guys and very happy to also hear from you guys later in the QA. How are you dealing with that? Maybe you have a different approach or you see similar problems.
And similar solutions, right? So today's work is really great work by my team that you see on the right. This is a photo that we took at ACL where he and I also met again. So this is really the great work of the team here that I'm just.
Honoured to present and we also presented some of that work at our annual SAP research retreat, which we just had before SEL in Munich. So you see a photo of that research workshop that we had in Munich at our labs on the left side. So yeah, all the credit goes to these people.
So benchmarks are a big topic now in AI. That's something that I think we as a field and as a community use to really measure the progress in AI. So whenever there's a

new model coming out, open AI or Google or somebody announces a new model, they always have this chart and they say, oh, we.
Are not top in MMLU or we are the top in the LLM arena and they have these big tables and and usually their model is now better than everyone else and everything and sometimes they even bolt their numbers if they're not the best. So so you have to really read them carefully and you also have to actually.
Be careful that which numbers they compare. I think sometimes they carrot pick them a little bit. But anyway, that's obviously something that people look at that also the labs look at and that generally I think people use as a form of guidance when they are looking at models and say basically how good.
This model right? So it's a new whatever GPT 5 now better than the previous GPT 4O. Should I maybe switch right? Should should we go from our previous GPT 4O deployment to GPT 5 or should we maybe go to whatever Gemini or or Mistral or Lamar or something else or to deepseek right?
And there's tons of these benchmarks, so it's not that we kind of invented this by any means. Here are some of the big and very popular benchmarks from academia. So the Helm benchmark from Stanford, that's one of our academic partners here that we work with on this topic as well.
They do a great job and we also use their framework extensively within our team. And then from our other academic partner, Berkeley, they have this very popular chatbot arena, what's now called LM Arena, where they.
Allow people to use models and you always get 2 answers without being told which model is which and which answer is from which model. And then users have to choose which one is better or whether it's a tie and they use these pairwise comparisons to then create a leaderboard. So these are super helpful and I I would. Already say if you're not doing anything with regards to benchmarks and eval then just Google this and find these websites and and whenever a new model comes out they probably will have them in their leaderboards pretty soon and this gives you a first indicator of is the model good or not and they actually have by now a lot a lot of.
Sub benchmarks. So I don't know in Helm they have a benchmark for Southeast Asian languages or for Thai or for Chinese or for reasoning, etcetera, etcetera, right. So they all constantly update them as well. There's a Helm scenario, there's a Helm logo. He actually already shows you how benchmarks are typically built. So typically you have a bunch of scenarios. Scenarios are basically.

Some kind of task with a data set and and a prompt, right? So maybe there's a scenario on on mass, right? And then there's mass questions and the mass answers or text summarization or sentiment analysis or reasoning, etc. And then you have all the models that are lined up here.
And basically you then feed every scenario into every model and then aggregate these individual numbers into an overall leaderboard. And and this is exactly what Helm does. Again, Elon and Marina has a slightly different approach which is pairwise comparisons from actual users, which is also very interesting.
And maybe a little bit more what people actually ask, but of course people ask a lot of stuff on this general open chatbot website. So that might again not be exactly what you are trying to solve in your team or your company. If you look at academia, this is actually a photo from from ACL that I took there that.
Resources and evals are now a huge topic for researchers as well. If you look at the bottom, there are the top subfields within ACL in all the different tiers, and resources and evals are really not there in the top things that people were researching on until 2024, and now this is the second.
This topic where people submit papers, so there's tons of papers benchmark for X, right? Or an X can be whatever Arabic or reasoning or planning. Because now we have these models that we even as researchers don't train ourselves, don't maybe understand even exactly how they were built and.
This proverbial black box that we now want to make sense of and the way to do it is basically to build a benchmark, right? If I want to understand what is a good in reasoning, I create a data set about reasoning and then feed it into all these models and measure the performance.
OK, so let me talk a little bit about what some of the work we've been doing and one of these is in the document AI space. We are we have been working on very actively for a long time. We have also been collaborating in this area with Appen before.
And one of the takeaways here is that when you are doing your benchmarks and when you're doing your evaluations, you just need to do that on your domain specific data sets in addition to looking at what the community and what the public leader boards and benchmarks do.
So the problem here is that we want to extract information from a business document. So here is an example of what an invoice typically looks like and if you OCR it's and so if you convert the image into text and you get something like.
The string there on the right, a document like an invoice typically has a header. This

is a stuffy on the top. These are fields that typically appear on the header, no surprise here, and that are usually appearing once on the document. Maybe there's bill to address, a ship to address.
Date the invoice number. Yeah, so that are things that basically there's a field and you want to build this block with a value, right? So you know you want to have the invoice number and you just want to extract this value and and just once. And then there's the line items. This is this table thing here and every row is a.
Line item and well these repeat usually. So you have the header here, quantity, description, price, each and amount, and then you want each row according to this schema that you basically get through a table, right? And when you OCR it, it looks like this. You see this just looks a bit more challenging. The layout is still something. Somewhat there, but this is a quite challenging task for even state-of-the-art AI models and there are benchmarks on OCR and all this kind of stuff, but they're often about various domains, including sometimes.
Some business documents, but these are generally hard to get because they're kind of confidential. People don't really put their business documents on the web for everyone to see. So they're often on things like maybe very simple things like receipts that people just scan with their phones, maybe on Street View kind of stuff, right? So you have pictures on the street and you want to.
I think signs or signs on the side of the road. And so what we did here in this research was actually we were interested in how good a state-of-the-art multimodal language model and vision models in doing this. So can we actually?
Skip even this part with the OCR and directly feed these documents now into a vision LLM and say here expect all these fields with a prompt at something like this, right? So kind of giving it a persona and being specific about the output format and instructions.
And then getting basically a nicely formatted structured schema like this one with maybe the delivery date, delivery note number and so on all expected. So these are some of the results. So we tried some of the state-of-the-art models at that time. From all the big labs and what is like maybe the takeaways and you need your own data set. So we had some in-house data sets from two different domains from delivery notes and payment advisors. I'll not go through all the numbers with respect to time here, but basically what we found.
Is that some of these models perform really, really good, even just with image modality. So in particular, some of the Google models here were very, very good

performing similar on image modality or even better on image modalities and.
Being fed to OCR text and I think for most of the models they were most robust if we feed them both the image and the OCR text. So having these multimodal capabilities can help to to get better extraction. But some of the models that I think on Google were were even better or on par with just the image modality.
So again, this is something that you can only find out by by measuring these things. And one of the important things is that you need your specific data sets for your task. So if you have already done a good job with regards to setting up your AI experiments, having annotated data of high quality and so on, this is setting you up very well for then running lots of benchmarks on LLM.
But this is still needed. So even in the age of LLMS, maybe maybe you don't need too much training data anymore, but you still need these data assets for develops.
Right. The other thing, usually when you have these numbers, you you probably get asked by your boss or maybe ask yourself how can you make it better, right? So these numbers are good, but then we get high accuracy and what we found here. So we did some work with using again LLM to.
Analyze the errors that the model is still making and then categorizing them and also here using the text and image modality and then grouping and clustering these together to then find.
Better prompts so we could automatically expand the prompts. So there's also now AI models that again can take prompts and optimize them by feeding them to an LLM and asking it to improve it based on the following mistakes that were made. So we were able to expand the prompt here, make it more specific.
In many ways, like more refining the format constraints, instructions and the schema descriptions, and we're able to get better accuracy. So develops are not just telling you where the models currently are, but they're also then.
Help you to identify where are problems and if you then systematically analyze these clusters, these that sets you up for improving your system, whether it's by prompt engineering or the alternative prompt optimization or any other means, right? So it can also be sometimes.
Simple things. Maybe just have a pre or post processing rule that takes care of some common mistakes and and sometimes when you look at the output of the models, it's really just small things that really can.
Degrade the scores of a specific model, sometimes adjust to checky or something. So maybe you say, hey, I want JSON like this and they say something like, Oh yeah, sure,

here's your JSON string, right? And then if you just look at whether this output format is correct, the model kind of gets probably 0 scores because it has this. Non well formed JSON string because it just kind of makes some helpful comments beforehand. So OK, next topic very much related. So first you need to do eval with domain specific data. You need to have then a way to analyze this, but there's actually also the question.

Of if we can now optimize the prompt, how does it change our eval, right? I already told you that if you optimize your prompt you can get better results and most of the benchmarks that we have in academia are very model centric. So basically they have these scenarios, they have the models.

And basically, it's a model's job to understand the task, and the model is just left as it is, right? Because I really want to see what is the performance of the model, how well can it follow instructions, how well can it follow the task if you're building a product or an application.

You basically now have two different things you can change. You can change your model and you can change your prompt. And in the end you just want to make it work as good as possible, right? So you. So there's no reason why you have to keep the prompt fixed like most of the benchmarks today too, and they just use the same prompt for every model and it's known in the community and everyone who builds with LLM knows that.

The models can be quite sensitive to the prompt and different labs have slightly different guidelines of how you should write the prompt and there's a lot of prompt engineering tips and so on. And there's now automatic tools as well that can improve prompts.

That's giving or just based on some examples.

So, so the question that we had is like, OK, if we would fix a prompt specific to every model, what would it do to the leaderboard, right? Would it change the the order and would it maybe change what we would then pick as the best model for our downstream applications?

So instead of doing the static evaluation, we have the static prompt that gets fixed for all models and then you eval data that you feed into the model and then get the metric and the leader board. What if we do the following? What if we have another small data set that an automatic prompt optimizer called PO here? That's another. AI model can take with initial prompt and then it can optimize the prompt to see how the score on the sample data set goes up. And then once it has the best prompt,

this now goes into the evaluation with the eval data that's separate from the sample data.
And then we do again the scoring of the responses with a metric and and calculates the leader board.
And um.
We did this with some standard academic benchmarks. So TSM 8K is a mass data set, open book QA and MMLU, which is about science and college grade.
College grade, science, history and economics questions, something like this. And then we did that with some internal data sets. Again, we just emphasized you need your own internal data if you really want to make sure this carries over to your final product. So we had something from digital assistance and.
Next to SQL generation and also this information extraction from documents use cases. And then we have two different setups. So we have one experimental setup where we only have the prompt and we try to optimize the prompt and then we have one.
Where there's a prompt and few shot examples. So these exemplers in in context learning and we try different optimizers, Textgrad, Meepro from DSpy and so on.
And this is work we present at ACL. So if you're interested, you can read the paper for more details.
So these are some of the results. It's a pretty busy slide, so I'll have to explain this a little bit. Let's look at the top row first. These are the instruction only prompt optimized experiments. So that's just a prompt. There's no few short examples. Then two set of bars here, one scenario.
And on the left one without any fades or something. These are the numbers for just the initial prompt, right? And every bar is 1 model. We anonymize the names of the models here. Higher is always better.
And you can see for GSMLK, for example, the orange model is the best, followed by the blue, by the red, and then the ones with the lines across right next to it. Those are those where the model has been optimized with a prompt optimizer.
And first of all, you see the scores are a little bit higher. This is what you would hope for, but also the order changes. So now the red model is the best and the orange one is only the second best and then the purple is the third best and the blue model that was the second best is not the worst.
So the order changes and overall you also get better performance, right? You get better performance overall, that's what you probably want and the order changes. So

you now probably deploy a different model than what you would have had if you just run the benchmark with the standard problem and we see basically the same result across the various.
Academic benchmarks, open book QA, MMLU and also our internal benchmarks that are here on the right.
And sometimes the differences can be quite significant. So if you look at this Eddy data set which is on information extraction, the orange model which is in the worst place or in 5th place suddenly jumps up to 2nd place. So there can be huge differences in the.
In the performance IT get.
And similar also if we run the same experiments with few shot examples, so these are the experiments at the bottom. Again, we always compare the initial prompt without few shot examples to an optimized prompt, but then also we have a setup with the.
Unoptimized prompt and few shot examples. Few shot examples typically help and then few shot examples plus optimization which has experiments with these dots here on top and generally optimization still helps. Few shot examples typically help and.
Again, the order of the model changes in most examples. Here's another visualization of the same results. It just shows how the rank order of the model changes. So for example, on GSM 8K, the model that was initially first, now a second, the model that was initially third is now a first, the model that was second is now.
Last, so there's quite a lot of crossing of these lines, which just means the order changes between not optimizing the prompt and optimizing the prompt, which again would mean in practice you would probably pick a different model if you run this with the optimizers and if you would not.
So we generally conclude you should probably optimize the prompts per model that requires a little bit more compute. It's a little bit more work, but these prompt optimizers can run automatically. You need a little bit more data now because.
Now you need this sample data set here. So basically that's the training data set for the optimizer. So whenever it changes the prompt, it needs to somewhat see is this prompt better than the last one. So it needs some data set to run against and then calculates a metric and say hmm.
Yeah, this worked. Or yeah, this one's a good, let's let's try something else. So you need a little bit more data, you need a little bit more compute, but you can pretty much automate this and in the end you probably make a better decision on which

models you should use.
And you get overall better performance, which is what you probably care about if you want to build something that users love and that works for them rather than just saying I'm really just interested in how the model is and I will not help the model so to say by.
Making the prompt fit for IT for this model specifically because I want to have a more clean maybe comparison model to model and so much models job to understand the prompt.
OK, OK and the same. So these are the the changes without few short examples. The same basically holds out for few short examples. Also here we see a lot of rank changes.
Although that were kind of leading on the leaderboard kind of go down, other models kind of go up. So the similar similar results.
If you look at a few examples, we could also show in the paper where this happens. So often it's about instruction following capabilities. Here are some examples from this information extraction case where the model here initially often outputs null values, so it.
Really writes now into the field when it's not present, but the instructions were to just leave them empty and make empty strings, which was in the initial instructions, but maybe not clear enough.
So here says return missing values as empty strings. It's there, but this is now more detailed in the automatically optimized prompt.
So it, for example, says additionally ensure that fields are never returned as now, right? And this changed the output and improved the instruction following capability and makes the optimized result match the the ground truth much better, right? And that's.
Can have very significant results. Of course, if you just manually inspected that, you might say, uh, come on, I mean now is also fine or so, but if you now have a downstream application that wants to use this Jason and it's not in the correct format, IT would probably break.
So this can be very important to make this outputs of the models useful.
OK, I'll be quick on this. So this is basically just analysis. You can look at the paper for details of how sensitive the different models at the are for the prompt optimizers. Basically different models are different. Maybe model B here has.
More significant changes is more sensitive to the prompt changes, so more red is

more sensitive and more blue is less sensitive and more stable. Model C, for example, is less sensitive, but IT also depends on the task, so some tasks seem to be.
More sensitive maybe this one, the Opilot help docs task and and others like open the QA show for most models less variations. So you can also look at that what kind of tasks are more sensitive to prompt changes or what models are more sensitive to prompt changes.
And then we also show some correlation analysis here. Basically we see that these rank changes are quite significant. So we basically compute Kendall style on those and we see the correlations between the non optimized and the non optimized rankings are quite low, which means.
Quite a lot of changes.
OK, yeah, for for details you can look at the ACL paper or happy to take questions later. I'll move on to another topic that's usually important top of mind for a lot of enterprises when they use AI, which is around safety and security.
We often hear people asking like, oh, what's happening with my data to use for training and so on. Of course, that one you cannot establish with events alone. You need some kind of enterprise agreement with your vendors or maybe deploy these models in your own infrastructure. But they're also of course concerned about safety topics, so.
The model should be not answering specific question. It should not tell you how to make bombs. All these things that we have seen that can happen with AI models that are going wrong. And there again are a lot of benchmarks also for safety.
Maybe calling out some of the great work that ML Commons is doing here and also the Helm benchmark from Stanford has a very comprehensive AR safety benchmark. But again, these are usually kind of static, so the way they usually work is they have. A taxonomy of different safety issues, hate speech, illegal weapons, something like that. And then they have prompts that are associated with these. So maybe tell me how I can make a gun with a 3D printer or how can I make a nuclear weapon, something like this.
And then they feed IT into the model at test under test and then they look at the responses and then there's a evaluator model that takes the responses and says is IT safe versus unsafe. I just the model says I'm not allowed to say this. This is probably safe. It says here are some instructions how to make nuclear bombs and this would probably be rated unsafe.
This is the setup that both of the patchwords use.

But therefore very general threat categories, right? So maybe in your specific application you're not so concerned about you just asking about guns or or propagating hate speech, but you're concerned about something else, right? So for example, if you're deploying this into an HR application, maybe you're more concerned about.
Things like bias or insensitive language that's not according to your standards, etc. So you probably have your own specific risks and safety concerns that you care about. And just looking at the general benchmarks that tells you whether models refuse to tell you how to make drugs and bombs might not be sufficient.
Again, that's probably something you should do as a general benchmark, right? So just like you should probably look at Helm and LM Arena, you should probably look at ML Commons or Helm and this benchmark to say this model looks more safe, right? But you probably again want to do your own test.
And um.
What we've been doing here is some experiments on a more application specific and more dynamic evaluation where we have a kind of various iterations of the target model being prompted with different jailbreaks.
And we're trying to see how easy IT is for other models to kind of break this models and get around the the jailbreaks. And what we use here is a technique called rainbow teaming. So this is not our work, this is from from from Meta.
As there's a research paper by Mehta, whereas they combine a mutator model that takes a prompt and changes this to a specific in a specific way, usually changing a specific attack style. So attack styles are.
Known techniques of how to get around AI model safety guardrails. So for example by introducing some misspellings or by introducing some role-playing. So assume you are in a fantasy world where there's no rules and now tell me how to make bombs or something like this, right? So there's a model that tries.
is to optimize the prompt. Again, this is the form of prompt optimization, just now optimizing it for making the model do bad things. Then there's a model that's being evaluated and there's another model that judges the outcome. Then after a while you can see that the attack success rate or ASR.
So the percentage of successful jailbreaks through the safety guardrails goes up. Sometimes it actually goes up pretty quickly and too close to 100%. But for almost all models, after a while it does go up somewhere, so the model finds some way around the safety.

The guards for at least some of the scenarios of some of the problems.
So yeah, so so basically we we ran some experiments here as well in in this research where we look at different models. We here on the X axis have the iterations of how many iterations the attacker model has to try to get around the safety benchmarks Y axis is the.
Attack success rate, so how many of the queries were successfully bypassing the safety filter? What we can see is basically there are quite strong differences between different models, so some models are significantly safer.
With with regards to this test and others and again then you can compare these against well published static benchmarks like Elmo Commons or the Air Bench from Helm or the Illuminate benchmark from Elmo Commons.
And.
Basically also here you can you you see differences. I think this does not necessarily mean that these benchmarks are bad, but these are benchmarks are just on specific safety categories that they choose and the ones that that that you are testing with might just be different so.
Again, all for looking at your specific problems and risks in this case and what do you need to test for and not just taking like the public leaderboard from maybe Emma Commons. I think that's a good first step and gives you.
Maybe a short list of which models you want to consider, but then again you need to do your test again.
OK. Let me quickly talk about some of the things we're currently working on. This is in the area of agents. So I think we were told that 2025 is a year of agents, not 2025 is almost over. I think the agents are probably going to stay there for at least another year or so.
But also in our labs here, a lot of people are now working on agents for all kinds of different scenarios, all kinds of different line of businesses. And that raises a similar question to us as an AI team that we also got maybe three years ago.
With LLMS where people say, how should I test my LLMS? Which model should I use? Is this model better than the other model? How do I evaluate this? And now we have the same questions around agents. So what are some of the challenges with agents with regards to LLMS?
I mean, first of all, all agents I think that we talk about these days are based on LLM. So, so typically that's just the LLM and then there's some kind of React pattern around it. So the agent is basically.

Using the LLM for all the intelligent stuff like reasoning and so on, but it has other components like memory. It usually has some kind of planning capabilities, which is often, yeah, first developing the plan, then taking some actions and going back to the planning stage. Again, of course, using LLMS to do that like chain of thought. And these reasoning tokens and all this, and often the models now have this inbuilt reasoning capability. So let's say have a reasoning mode that you can switch on that outputs reasoning tokens. And then one of the big differences to just using LLM's and just evaluating LLM's is the.
Fact that agents typically have tools, so it will be.
To be really useful, the agents need to take some actions if they want to maybe assist you in your travel booking or in paying an invoice, or in solving some problem in supply chain or whatever your agents are supposed to do.
So they will have some kind of functions or APIs that they can call. And now the question is a little bit how do you evaluate this whole system? So this is more complex than just doing the LLM evaluation. So if I just quickly go back for the.
LM regulation.
This is a pretty good set up already, right? So you just have some inputs from your data set, from the evaluation data set, you put into the model, you get your output, you have a metric, you get a score and that's a little bit more complicated now that you have this agent because the agent is not just.
An input output response interaction model. But now the model, the agent would do some reasoning internally. It would call some tools. It might take several iterations to come up with a solution or a plan and.
Then finally, the answer that the agent returns to you or the action that it takes might be right or might be wrong, and you probably want to understand a little bit more where this agent also goes wrong. So again, you can not just say, Oh yeah, the agent works.
Whatever, 50% of the time or 60% of the time, but also um.
See how you can make it better, right? So, so it's overall the setup is more complex and that I think creates some new interesting challenges also for benchmarking and evaluation.
And basically there are different agents of different complexity. So the most simple agents that are just having a single turn static conversations are very similar to LLM's and you can basically use the same techniques that you used for LLM evaluation.
And for evaluating this agent, so so maybe a very simple agent, you just and ask a

question, you get one response and that's the whole interaction of the user agent interaction.
But typically these agents will have multiple turn, multi turn conversations with the user. So it's not just input, output and then I score the output against the expected answer, but.
And you have a few more rounds of this and then for example, the question is do you want to maybe cut this conversation at some point if you feel there's not going anywhere and you're just wasting time and and tokens on this conversation?
And for multi-tron conversation, again, you can evaluate this in an aesthetic way. So basically you have a redefined set of user queries and they always follow the same order. Maybe that's a certain workflow where the user always asked.
First for A and then for B and then for C and no matter what the agent says, the next user interaction will be the same. Then finally, there's a dynamic multi turn setting where the interactions of the user.
Change depending on what the agent says, and this is the most complex one where now you need to replace your user with another agent that simulates the user during test time because you want to run the test automatically and not always have a user who needs to sit there in front of the computer to.
interact with the agent. But you want to make this automatic and you want to make it repeatable and reproducible. So So you need to now engineer another agent that is instructed to act like the user in this interaction.
And again, this makes the whole evaluation more dynamic, a little bit like what we have seen in the safety test, where it's not just here's an input, produce an output as for the output, but you have this more dynamic interaction between the simulated user and the agent under test.
And in the end you wanna be able to say now how good is this agent under test the guy on the right side with regards to fulfilling the task?
Right. So with that, I pretty much threw the content I wanted to cover some of the takeaways here from from the different piece of work that I talked about. So evaluations are super helpful and important in this area of AI to help us understand. And how models perform in general, but then also on your enterprise tasks. So they are an essential part of your activities in your AI team. Public benchmarks are great, but running your own benchmarks in house and your own eval on domain specific benchmark is essential. Otherwise you're essentially flying blind and you don't really know.

All these models work for you, whether the new model update would be better, whether you should switch to this other model that just came out. And this gives you like facts to talk about because sometimes people also just have opinions, honestly. So we have definitely had other conversations where people say, oh, I saw.
On whatever Twitter or public media, people say this model is really good or this model is really good and I played around with this over the weekend and I think this is better and this is not engineering, this is not scientific. So you need your own benchmarks.
Yeah, and new models and non model version get released all the time. Feels like every few weeks there's a new model and people say, oh, breakthrough, blah blah blah.
And yeah, again, you need benchmarks and eval to put some numbers behind this. You need this to avoid regression. So eventually you will have to change from one model to the other because the models get deprecated after a couple of months, so they won't be around for years and don't have this.
Maybe enterprise support that you can now use them for the next 10 years without any changes. So you will have to change and just like you do have regression tests and all these things for your normal software development, you want to have your evals and benchmarks that you can confidently switch from one version to another. Without having.
Now, no idea whether this will work or how well will this work. You cannot fully guarantee that the new model will be exactly like the previous one because you will not have 100% test coverage. But I think this gives you a much, much better setup and much better confidence to making these decisions.
And of course you want to take advantage of improved model versions as well, right? So when a new model comes out or another vendor has a new model and that's better, you want to know, you want to know whether that's better and you want to take advantage of that raising your application performance with better models instead of just being stuck on.
Whatever models you you started with when when you built that thing and yeah you need to cover different dimensions. So evals are kind of multidimensional or there should be multidimensional. You want to care about accuracy. Is this response good? Does it?
If the correct answer, but you also want to care about safety and then you might have very application specific risks. Again, like if you're building a HR application, you

probably have to look into things like bias and discrimination and and so on, maybe even for mandatory compliance reason.
While if you're building this invoice scanning solution for example that I talked about, bias is maybe not the top priority of there, but might be more really the accuracy and reliability of extraction result when you put these numbers into your financial system. And you probably don't want to post wrong invoice amounts, but IS is maybe not so common in whatever is in your line item. It's not impossible, but maybe not the top concern. So again, this needs to be application specific.
And finally, the whole benchmarking exercise and and all this evaluation topic kind of moves so to say up the stack a little bit. So I think there's still tons of work on model evals, but now there's a lot of research on agent evals and then of course.
In the end you also have to do your application specific evaluation and tests all the way from a from a user perspective to really understand whether the whole application works. So think a little bit like the typical classical test pyramid that you would have. You have unit test and then you have integration test and system test. And user acceptance test all the way at the top. So you basically need to think about that in your evals and benchmarks as well. You need to test the models, then you test the agents or whatever you build on top and ultimately the application end to end. Right. Yeah, that's kind of how we look at that and how we work in our team and I'm happy to take some questions. Thanks very much.

Si Chen 50:14

Thanks, Daniel. I think that was super interesting people. So we can feel free to put questions into the Q&A, but I'm gonna just start with one of my questions. And I think for me it was really interesting to see that there was actually a lot of movement in the relative rankings when you used prompt.
Optimized prompts versus the model centric prompts, right? And I think that you know for me to see a model that performs really well on a public benchmark actually kind of be the worst performing model once you've actually optimized the prompts based on your application. You know, I think that's the kind of information that maybe a lot of people aren't necessarily aware of when.
Understanding or looking into the benchmarks, do you think there'll be any changes to how benchmarks are built to actually address the fact that they're not necessarily optimized against the models?

Dahlmeier, Daniel 51:07
Yeah, I mean we've discussed this with some of the people from Stanford Town project and we, yeah, I mean we we are thinking maybe we should build this into this framework so you could kind of turn a switch and and you get the optimized benchmarks I think.
Yeah, I think that that that could make sense, but I think both have value. It's not that one is necessarily bad and the other one is always better. So if if you're really saying, OK, I'm interested in the models, all the models have the same conditions, they have the same chance, they get the same prompt, everything equal.


Si Chen 51:29
Mm.

Dahlmeier, Daniel 51:41
And apple to apple comparison of model A against model B, you can you can argue why you don't want to change the prompt, right? Not just because it's maybe tedious and more work, but but you want to say, OK, everyone had the same instructions, everyone had the same starting point.


Si Chen 51:48
OK.

Dahlmeier, Daniel 51:56
I just really want to have a fair comparison of whether this model is better. I don't want to tweak IT for everything. But if you're saying, look, I mean, I just want to make this work as good as possible and now I have two variables I can change. Why don't I optimize them together to to make the overall best system?
So, so both I think have value but but but yes, I think there's some benefit in in tweaking the prompt and and there are of course also other like I I talked about the LM arena, right. So this is also super interesting that they have this more.
Based on real users feedback, so so there's a lot of benchmarks and and I wouldn't be surprised if also some of them have now in the future some option where you can say run optimization before if else or something like this so.

Si Chen 52:34
Mm.
It is.
Mhm.
Mhm.

Dahlmeier, Daniel 52:51
Yeah, but both are, both are. It's not that one is wrong and that's not our point, but it's just like measuring slightly different things.

Si Chen 52:52
Yeah, I.
Mhm.
Yeah, I think it's interesting to because obviously there's different trade-offs and as you mentioned, I think it makes sense depending on what your purpose for measuring actually is. I think there's a lot of interest in the prompt optimizer and I've got another question here which is around the training data for the prompt optimizer.

Dahlmeier, Daniel 53:18
Mhm, mhm.

Si Chen 53:20
So the question is, does the training data typically have examples of optimizing the prompts for multiple models? For example, the way to optimize a prompt for ChatGPT may be different to the way to optimize a prompt for Grok, Gemini, or other models. So can you expand a little bit more on the?
Kind of how to do the how to create that training data.

Dahlmeier, Daniel 53:43
Yeah, sure. So, so I mean the the training data for the optimizer, what is sample data on this figure here that's basically similar to EFL data, so that's the same format, the same scenario.

Si Chen 53:56

Yes.

DD **Dahlmeier, Daniel** 53:59

So I don't know if it's text summarization, then there would be a prompt that says please summarize the text and then there's some long text and the short text to expect and and the metrics. You also need to metric that measures how good something is and the prompt optimize. That's another AI model. It's some kind of. Gradient free optimizer. Sometimes it's an LLM, so it's LLM base and LM gets instructions. Here it's a prompt. Um.

How can you make the prompt better? I mean, not not exactly like this, but but and sometimes it's some other gradient free optimizer techniques. There are different techniques. I mean we used here in this experiment.

Si Chen 54:27

Yeah, OK.

DD **Dahlmeier, Daniel** 54:40

Me Pro, which is implemented in D spy. That's a open source package from for building with LLM from from Stanford and Techsgrad, another project from Stanford. We also open sourced D spy based prompt optimizer together with Meta last year at Lamacon. You can find this on GitHub.

Si Chen 54:55

Si.

DD **Dahlmeier, Daniel** 54:57

The different ones we found that these were.

Yeah, I mean, there's some differences. You can try different optimizers. It's a good question whether one optimizer specifically good for a specific model.

Let me think of this. I think text grad, so, so probably, but I don't know whether it's very systematically. I think text grad was pretty consistent across different models. There were definitely some models that were.

Or rather, there's some optimizers that only work with specific models. So for

example, Google has APD prompt document called APD, and that's part of GCP now, and IT only works for Gemini. So or rather, IT only works towards Gemini. I think you can only you can take.

Si Chen 55:40

No.

Yeah.

OK.

DD **Dahlmeier, Daniel** 55:47

any model, but then you can only you can act any prompt. So maybe you started with building with OpenAI and then you have a prompt and then you can optimize it, but only for Gemini models. So obviously they want to get people to come to Gemini. So yeah, I think it's possible that certain optimizers work better with a specific model.

Si Chen 55:49

No.

Mhm.

DD **Dahlmeier, Daniel** 56:07

But I don't think there was a huge difference. So it's probably not. I mean, maybe you can squeeze out a little bit more performance by running different optimizers and seeing which works best that that's something you can do. But I don't think there was a very systematic difference that this optimizer works specifically well for this model, except for those.

Si Chen 56:11

H.

Hmm.

Mhm.

DD **Dahlmeier, Daniel** 56:27

Where the people who built the optimizer force you to only use IT for one model

because that's how they built their platform. A good question though, maybe there's some research that you could do, but yeah.

Si Chen 56:31

Hmm.

Yeah, yeah.

Dahlmeier, Daniel 56:42

Mm.

Si Chen 56:43

So far, obviously we can see the big improvements with the prompt optimizer, but whether or not there's further improvements with customizing your prompt optimizers for models could be could be worth looking into. There's a couple of questions here which is related to red teaming and I guess model safety is always very top of mind for.

Dahlmeier, Daniel 56:48

Mhm.

Yeah.

Si Chen 57:02

And for anyone who's building enterprise AI, I think the first question was whether or not the work that's been done so far has been on multimodal red teaming of the models or have they just been primarily text based prompts?

Dahlmeier, Daniel 57:04

Mhm, mhm.

We have worked with text based prompts in our work here, but there are some red teaming safety benchmarks that also use multimodal.

Yeah, I I I don't remember exact reference, but there's definitely some work where maybe you have a text-based instructions and the model said no, I can't do this. And then you take an image of this and feed the image and then model said yeah, sure, or or make ASCII art out of it or all kinds of funny stuff. They definitely work on this. Um, but uh, in our work we have been working with text based prompts.

Si Chen 57:56

OK, I might just shift because I'm conscious we've got about 5 minutes left to I guess some of the more general questions. There's one in here in the chat which is related to.

I've often heard that when evaluating at the application level, it's better not to start directly with quantitative benchmarks, but instead begin with qualitative evaluation, e.g. for example, error analysis through domain experts. Um, you know, what are your thoughts on the importance of starting with this qualitative error analysis? Um. That that you could provide.

Dahlmeier, Daniel 58:33

Yeah, I mean.

Si Chen 58:33

I mean, do you agree?

Dahlmeier, Daniel 58:38

Yeah, I mean, I mean, so so if you if you're just starting out, right, so, so I mean that's a good thing with the LLM and and all this thing is it's it's really easy to start, right. So typically when you when you're building the classical machine learning model first you need to.

Oh, you need training data and maybe thousands of those. I need to label them and before you get any results it it's a lot of effort and you know you can just have this whatever playground prompt editor thing and you just try out something. So actually it's totally fine to just go to your chat bot.

Playground or whatever and just play around with your few models, right? So just just kind of write a prompt, maybe play around with a few different options or some prompt engineering and just try a few models and maybe you already see that this model looks a bit better or maybe this is too verbose or and that gives you maybe some.

Si Chen 59:13

Yeah.

Dahlmeier, Daniel 59:26

Vibes right now. It's all about the vibes and and maybe that's totally fine, right? But I think you you still want some quantitative and repeatable experiments, especially when the new models comes out.

Um, So what we have seen also when when teams, um when our AI unit started building.

Maybe you don't have test data at the beginning, right? So so I mean that data is is is hard. It's hard to get for specific domains like enterprise applications. Of course you can take these these these public benchmarks, but they're not so interesting for your task and maybe before you have such a data set and hundreds of examples in different annotated.

Si Chen 59:53

Mhm.

Mhm.

Dahlmeier, Daniel 1:00:07

That that takes a lot of effort and maybe you it's totally fine to just say I just have a few samples. I just throw them in here. I I get some feeling whether this model is better than this model. Yes, I think that's fine. But then you probably want to run this repeatedly and whenever new model comes out. So when you just do this manually and qualitatively.

Si Chen 1:00:18

Mm.

Dahlmeier, Daniel 1:00:26

Um, we have seen teams that even do this very well, but then the problem is.

When a new model comes out, your old annotators and and domain experts, maybe they're not around anymore, or it's too much effort to come back every few weeks or every few months and say, hey, can you judge this again because now there's a new G55 and we want to test it and they might not be consistent, maybe they're not around anymore, then you can't compare it anymore. So this is the problem with this. They don't age very well. They're not repeatable.

And and what we see when we now, so we run our automatic benchmark whenever any model comes out. So the the platform teams comes to us to say, hey, here's this new model, this is coming out soon, can you run the eval so we know whether we should put it on the platform and we have done everything, we have everything in place for the release.

Si Chen 1:01:01
Mm-hmm.

DD **Dahlmeier, Daniel** 1:01:11
And the benefit of having this automatic is we connect the model, we run the eval, the leaderboard gets updated and everyone can always see, oh, where's the new model? So yeah, I think it's OK. You probably still want some like just in in software testing, some user acceptance testing at the top, right? And and also this is.
But always 100% repeatable, but it's also expensive and slow, so I think you you want to have both in the end, but just playing around with some models, especially if you don't have the data yet and it still takes up time to collect it and to rate it and annotate it.

Si Chen 1:01:42
M.

DD **Dahlmeier, Daniel** 1:01:46
Totally fine, yeah.

Si Chen 1:01:48
Yeah. And I mean, IT sounds like IT can vary. You know, the answer can vary depending on what phase you're in. And I think that, you know, we've done a lot of work in the past where we've supported a lot of the upfront qualitative evaluation that then forms test sets that can be used on a more repeatable basis. I think that's actually.

DD **Dahlmeier, Daniel** 1:01:57
Yep.
Yeah, enter.

Mm.
Yeah, and you can combine them as well, right? So you can look at maybe your your logs of your chatbot or whatever and then you can you can kind of text things like OK, this is instruction following and this is whatever tool calls and and then you can again use things like this way and.

Si Chen 1:02:09
Yeah.

DD **Dahlmeier, Daniel** 1:02:22
I mean this is was done fully automatically, but you could do this also like human in the loop, right? You can have some humans that label some stuff and then you have an LM help you summarize it and cluster it and and this can be super powerful. There are some tools even for this now, but you can also do this basically.

Si Chen 1:02:38
Mhm.

DD **Dahlmeier, Daniel** 1:02:39
With an axle if nothing else and collect these things and this can be be very very powerful and it's also something that domain and product experts can do. So you can have a product manager do this.

Si Chen 1:02:47
Mm.

DD **Dahlmeier, Daniel** 1:02:55
So you don't necessarily just have to be a data scientist to look at this and label it and then have an LLM maybe find patterns in this or cluster them for you. Yeah, so I think long answer, but short answer is yes, that's fine, yeah.

Si Chen 1:03:09
I'm, I'm conscious of time. I'm just going to try and squeeze one more question just because I think agents is obviously like you said was the kind of buzzword for 2025,

but I think also 2026. So I think there's a question in here around for agent evaluations, what are some of the top factors?

Dahlmeier, Daniel 1:03:15
Sure.
Yeah.

Si Chen 1:03:29
That you would typically consider at at an enterprise like SAP outside of the, you know, the typical performance metrics like safety costs, like what would you prioritize as being the most important factors for agent evaluations?

Dahlmeier, Daniel 1:03:45
Yeah. So I mean if you look at the agent, if you look at benchmarks for agents and and generally the research on agents is that while I think there's a huge recyclement and there's a lot of.
Progress there as well. This is still extremely challenging, so agents can often do simpler tasks, but they often still struggle with more complex tasks. There's certain domains where we've seen a lot of progress, like software engineering.
And so on. But with, yeah, I mean, so, so I think the the main metrics that we look at is still success rate. So does it work? I think that's ultimately what people care about, right? So if I give this task to my agent, will, will it?
Do the job or not, right? If it's not doing it and I try three times and it's not doing it, then I might as well do it myself. Um, so I think we're still our um.

Si Chen 1:04:41
2.

Dahlmeier, Daniel 1:04:42
Yeah. I mean, of course safety cost all of this is important. In the end, you need all of those to have a successful product. But what we see that because agents are often not able to complete the task in in many more complex and and challenging settings, maybe enterprise tasks and so on, what I think the literature is doing.
And what we're also doing is to look at also more partial success criteria. So for example, there's a metric called progress rate that we also use instead of just final

success rate. Because if a model, I mean, maybe you have two models and both didn't succeed in doing the task, but one got 80% of the way and one only got 10%.

Si Chen 1:05:21
So.

Dahlmeier, Daniel 1:05:22
Of the way, the one that's 80% of the way is probably the better one to focus on. So you probably want to have something more fine granular than just did it work or not. And the other thing is you want to look at metrics that help you to improve the agent, right? Because you will not have an agent that always works, probably also not the agent that never works.
But you will have an agent that sometimes works and you want to make it work better. So it makes sense to look at what we do is we look at metrics that help us to analyze where the problems are and how to improve it. So for example, we have metrics around tool usage, right? So all these things.

Si Chen 1:05:55
OK.

Dahlmeier, Daniel 1:06:00
So it's a problem with the tools. Does it not know how to? Does it not discover the tools? Does it not find the right tools? Does it find the right tools but not formulate the right payload? Or is it the planning? Or is it the memory? So you probably want to have different metrics that help you to kind of debug and analyze what's going wrong so you can make it better.

Si Chen 1:06:03
Um.

Dahlmeier, Daniel 1:06:19
Better. Yep.

Si Chen 1:06:21
All right. I'm conscious we've run a few minutes over time, but thank you, Daniel, for

sharing all your expertise. Really interesting session. I think obviously people had a lot of interesting questions. We weren't able to answer everyone's questions, but you know, if we want to continue the conversation, feel free to connect with us on LinkedIn.
Mail us or just reach out through our website. We'll share the webinar recording after this. So again, just a huge thank you to Daniel for for his time today and I think we all, you know, learnt a lot from this. So thanks all.

Dahlmeier, Daniel 1:06:54
Thanks again for having me. Have a great day everyone or great evening wherever you are.

Si Chen 1:06:57
OK.
All right.

stopped transcription