Getting the Most Out of Your LLM Experiments

48 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=IFCDVTL6Z1Y
https://www.youtube.com/watch?v=IfcDvtl6Z1Y

SUMMARY

Thomas, a machine learning engineer at Weights & Biases, discusses the importance of reproducibility in machine learning experimentation and how to effectively utilize tools for tracking experiments. He emphasizes the challenges faced in managing hyperparameters, datasets, and evaluations, and showcases various integrations and features of Weights & Biases that facilitate better experimentation practices.

- *Reproducibility is crucial in machine learning, allowing for better tracking and understanding of experiments.*
- *Weights & Biases provides tools for experiment tracking, including integrations with popular frameworks like Hugging Face and Axle.*
- *Users can log hyperparameters, datasets, and metrics to create comprehensive dashboards for analysis.*
- *Collaboration and community engagement are encouraged through shared fine-tuning experiments and discussions on Discord.*
- *The importance of inspecting model outputs and not solely relying on loss metrics for evaluating model performance is highlighted.*
- *New features, such as the Weave tool, help trace inputs and outputs for models, enhancing the evaluation process.*
- *Thomas shares insights from various user experiments, showcasing the versatility of Weights & Biases in different machine learning tasks.*
- *The session concludes with an invitation to explore the tools and engage with the community for further learning and sharing of best practices.*

so hello everyone thank you for having me here I'm Thomas I'm machine learning engineer at weights and biases I'm based in France so let me tell you a little bit about myself and what we're going to talk today we're going to talk about reproducibility and how get the most out of your experimentation I am a machine learning at the yeah growth part of weights and biases I'm mostly working on fun I became kind of the fine-tuning

guy I want to be I'm working internally with llms and externally trying to help customers get the most out of the fine tunes and all the problematics you have been showing during the course actually they're very real a customer basis I also contribute like a lot on the open source ecosystem I I built some Integrations I I contribute to open part of weights and biases like the examples repo and the Axel integration some of the highing face Transformer

spots I recently collaborated with the Tor tune to have a weights and integration in there and yeah I also like work with our new prodog that will'll show at the end that has M integration already built in previously I was working more in the traditional ml part I was working with time series data so maybe some of you would relate

with geospatial data some computer vision with satellite imagry to actually create malls to forecast cast renewable energy

production so some electrical engineering with the the data science data analysis and then like deep learning actually I'm also a fast a attend yeah student from multiple versions I collaborated back in the days in both fast 1 and two and and I I really like like this open source and Discord community that has been formed so yeah I I want to just start by saying that keeping track of everything you do on this field is complicated in computer science is

generally complicated to take care of everything but here we have hyperparameter data sets now we have prompts and we have to run evals and like I feel like this guy multiple times like I need to bring my doer image I I forget how to run that put on a computer and maybe a switch provider so like you need to be like a Swiss knife here and I learn a lot of different toolings to be able to like to Port your

code to different envir ments so we at weight and biases are company that started like creating tools for experiment tracking and as you have seen during the course there are multiple users haml showcased the Axel integration wing also came and and show how to use it I'm not going to most of you know what weight Andis is so I'm going to just point out that we have Axel integration and most of the fine tunes these days also happen on the

hiding pH trainer so we also have a very simple integration there it means that you just have to pass a flag and you will get your dashboards these beautiful dashboards with the metrix log to the central repository of metrix so instead of like just talking about that I'm going to show some of the work you have been doing as someone pointed on the Discord I have been trying to help people around and how do I do that

because people share their fine tuning experiments and then I can like sneak peek and try to help them out the metric what they are logging maybe something is failing so we have this reproducibility B Team when you use a product like this so just to show sh out some some of the users here D show is trying to like his first fine tune and using like alpaka data set and trying to run L on Up of that and yeah and haml here set up

like a a fine-tune channel subchannel that you have like threads and people can dig in and share their experience on their fine tuning Journey so I've been like sneaking in there and trying to see what people are doing so we can we can actually click and and see what Damon is doing here and yeah he's he's actually performing some some different fine tunings here maybe twigging some parameters and as you can see he has run multiple experiments and he has like

highlighted three of those for us like this little eye let's look let's you toggle on and off different experiments as as you toggle the runs like pop in and pop out it's very Dynamic and we can actually dive into the one of the experiments and and maybe the metrics are organized by train and eval here the learning rate the loss is kind of going down there and we can check on the overview Tab and see like oh he's using

Axel all so this was run with with Axel CLI train and there's like a a yaml file with the config and we actually grabb the config automatically for you so here you the the parameters he use for this fun I think config is already also in the files now too yeah yeah it's it's actually log here like there's an artifact here where we have the actual Raw AML so you can pull that and and run the exact same experiments he's doing so

this is the actual confli not bars like raw yaml I think that's really useful because you want to actually pull the yaml and then like rerun that experiment and maybe tweak something so having the the actual yaml as a first class citizen here is is very useful so yeah he he has run a multiple experiments that's really cool so we can actually see what he is doing how the Rands are performing so so to come back to the slides yeah another

cool project is Zach he was trying to do some self instruct here and he was like sharing some knowledge on this like rocking points kind of at the end of the eok we say that maybe memorize some of the data and that's why we have like this deeps on the loss he was trying to improve his experimentation tweaking some parameters here and there there's a link to the the actual dashboard if you want to check it I feel this is

really interactive or what was interesting about this experiment I think it was he was actually tricking some stuff on the on the trainer himself I don't know if if Zach can tell us more about this experimentation but he was he has multiple projects here one one of the ones I link here is this one but I think there the screenshots are not from this project exactly but yeah we can see that actually he's running on with a lot of

like override on the fly with he's probably using the trainer here straight trainer and he's like running on an 8 100s you can see that yeah it actually chained pretty fast so it's yeah we can yeah he was actually talking I think some of these runs crash so we go to the runs View and see on table yeah some of them fail or crash proba he Qui kill them before finishing yeah the long runs are like

six hours here check one of those like this one yeah he was trying to like tweak the parameters to actually see if he could avoid this small like as you can see this one has like way less but I'm curious if he like actually no the learning rate is still pretty reasonable like because one technique you could do is like put a super low learning rate but that means your M like learns slower so it's not always like

the Silver Bullet here so yeah that's some of the experiments like Zach right there's there's a big thread there are these PL are there is this this is related to the like the answer AI post about yeah this relates to the answer yeah jono jono and Jeremy like did some experimentation about like trying to memorize one badge and and actually seeing this but this is like a longer Kore experiment he was doing yeah she was trying to actually I don't exactly

know what he was doing with the prompt here but we can ask him on the thread I encourage you to be able to point other cool experiments we have seen is that yeah someone is trying to actually teach kind of a better function calling to M trial this project is not public so probably there's like accurated handmade data set that people don't want to share so to actually be able to share the the configurations he was using he

created like a something we call a reporting weights and biases is that his showcasing only a pod a sneit of what he wants to show here so we have like the actual config I think he was running into issues with the Axel all config so he went some debugging so he pasted the config here so and some of the training metrics I'm curious about this because there has been like other models like know technium train like Hermes pro has some

function calling functionalities before Mistral added function calling to the instruction instruction models and there's like a lot of forms you could code the data set to to actually support function calling maybe adding special tokens or not so yeah this is a project to take keep an eye on it and maybe if they push the final checkpoint we could actually test it and and Benchmark against the other malls that have function calling I don't know what you say how but like yeah now this is

he's not running the mistal 87b as you can see here is running from the Mistral 7B but I think this was like some experiment naming that he put at some point I'm I'm curious how good a seven could be at function calling without losing too much like knowledge on the previous data you showed him like you need to show a lot a lot of function calls to be able to actually learn that so and yeah what other products we have

this one was pretty cool because he actually Shar like a a project we build internally at weights and biases this is a user that has been like training a lot of Japanese and yeah trying to get better bilingual mods in Japanese have I have worked on Japanese data previously with gbd4 and it's good but it's not always perfect it's way worse than than other languages like Latin languages are way better sometimes it's like it messes like when you have like

three subjects in a in a sentence it kind of makes them app so yeah there's have been efforts like like na in Korea and this to actually have a specific language model for for Japanese Korean so this is really cool so he show he link sorry link like the leatherboard we are trying to build with a Korean team and Korean Japanese team to actually get a good scoring of the capability for those malls on on those

languages so you can you can check it out and he's a very seasoned fine tuner so he has a lot of experience and yeah of course during the course as multiple persons have show like weights and Vis dashboard or talk about like logging yeah jono is like a user waiz for well sopia showed that they build an integration on the fine tuning API to actually enable you to log your your experience there and HL has showed some

Integrations with Axel and how you can log your metrics there and other people outside of this scor like carp uses use it on nanog gbt Mark is using it Wing use it a lot to debugging all the axle improvment he's doing Jeremy has shown some metrics probably it was not Jeremy that built that dashboard but I know Jon and and Benjamin use this internally and yeah maer that he's in the course he's also like pushing a lot of of

checkpoints and good mods to the Hing face Hub and he's also using weights and biases to to keep track of the experimentation so I want to show you like another dashboard something I built recently that it's like a bigger project that maybe you could relate to that this is like an experimentation we did with with a couple of person actually from Discord we created an open source group jono was was there at the beginning then he like helped it out at the

beginning to actually organize this experimentation and we were doing some ablation studies on Mistral 7B so we were removing layers and retraining them all on two phases like sft and DPO to get them all back to talk because once you remove layers because we were removing like 75% of the layers the model was broken so in this project you see like there is a way more like complex dashboard in the sense that yeah we're grouping the runs that's something

you could use like if you have brands that relate to something maybe grouping is useful so in this case we have the same initial model like one St one stage two stage and then the evolves another cool thing you could do here it's it's actually having different views of the project so this is like the general dashboard that is very kind of meaningless because it's very bloated so I have like a special view that organize the dashboard in a way to just check the

first stage of the fine tuning last screen one second sorry what is the going back to the previous screen you're showing oh that one yeah what is this like shaded area what is that yeah because when you group like it takes the it Shades the the full curves it takes the average of the corves so it doesn't make much sense here so this is like kind of a a dashboard that is not very useful because it's trying to compare like the

train laws of two different training recipes that you cannot compare actually on the same dashboard that's why in this project like there's like the default view but it's not very meaningful like I have an sft view that only filters out like we're using like a Rex here and filtering out only the sft runs and then I can compare the training losses together I think of so I think of hugging face tools as being really nice for these abl studies I don't know

whether you get to see very closely what people are doing when they're doing these ablation studies but if so is there anything that you've seen that jumps out and now when you talk to new customers you're like oh I saw this pattern and it would be like a really cool thing for you to do in the future or is there like any workflows that you think or insights for people who are doing the doing ablation studies that

you think other people should know yeah this project has a lot of context like I can I can actually put the link on the Discord it has like a pres of one hour it's actually at the time we built this this was when tiny llamas started training like a small llama before before five before before this small mall so the idea the initial idea of this was let's create a really small Mall from the big mall not

not training from scratch so there is like papers on that like sheer Lama that actually yeah do like smart ways of pruning them all this is like basic just removing layers completely we actually Jeremy at the beginning suggested some of these strategies and and yeah he said like just remove the full layers like it doesn't matter if it's like not that deep and it's shallow so we started playing with that and then we kind of abandoned the project and

then we came back later we got some compute and and we were like okay let's let's see if we can get a small Mall because then drafters came out like so we could use a small mall that it's a similar as possible as the big one as a drafter so we were like okay let's try to make a 2B or 1.5b for the mistal instruct but we don't have the recipe for mistal instruct so here we use the

sephi recipe that at the time was one of the best like instruction tuning recipes so that's why it has two stages like sft and DPO this is actually applying the same sey recipe so you destroy them all and instead of like doing continuous pre-training you recover them all just straight with instruction tuning and it works like the the final mall has a reasonable score of course not as great as a big mall but it we put like a super

limited budget of 1 billion tokens that's very very low and we were able to have a mall that drafted the big mall and they kind of work and a fun fact like during the mistal hackathon another team was trying to build like an embedding Mall from the Mistral 7B and I talked to the team who was like oh but I did this you can try the prune mall maybe it's a good and better and they were trying to do something really fast

and actually the mall is very good and better so yeah there there's multiple usage of things like this of course it was like a research project kind of not targeting like deploying this mall but yeah it was good inside like we were not expecting actually to recover them all so fast with instruction tuning and yeah also we try a lot of stuff automatically here so like when you push the upated like we were quick we create a script that bladed layers

and then you push the checkpoint as to weights and biases and we trigger out automatically all the steps like the supervis fine tuning the DPO phase and then which trigger eval harness that that's like the framework from a lutheri that runs the evaluations so I also have like a nal's dashboard here that is like organized for evils so we have like the standard eval harness metric that will come back to the dashboard so as you can see the big Ms are way better

and like the small malls are slightly worse no half as good but reasonably good for all that couldn't speak English before training so yeah that that was like what I want to show you here this is like a more complex workspace that has all the bells and whistle from wees that I will not dive here but we can discuss this later if someone pings me and yeah that's what I want to show you and maybe like yeah another trick I want to show

you here is that some people like in a dashboard like this one some some things that people don't know that they can do is that sometimes you have trouble like inspecting your runs here and instead of showing you a project of mine that has it I can show you in this project that's from that on yeah you can see this runs View and most of the time when I'm interested like in a metric here I will actually Bay some

of these metrics maybe I don't know I want to see this parameter and you have the spin column option that will let you like bring the column to the left so you can organize your workspace and the Metro is going to be always bus ible and another cool tool I may you may not know I'm not going to show you everything but there's like I also find that for you guys doing fine tuning experiments on this course the run compar is very

useful that actually yeah shows you the differ hyper parameters for the for all the experiments you have done and actually you can toggle def only so it's only showing you the difference between the runs you have toggle on and off so if I toggle that on it will like reren there with that so in a bigger project maybe this one like I have a big project I did with Morgan a couple months ago with fun Mixr when Mixr got

released so we were using this run compar to be sure that we were not messing up and the the version of the axle all was changing the Transformer underlying version was changing we were like pushing fixes here and there because like there was not an an actual mixt fine tuning recipe there established it was too early so we were looking at at the difference using this run compar table and like so that that may be useful if you're doing a lot of

experiments you activate this comparison table I haven't seen that one before is it oh it's it's just if you are in any project like this you add a panel here and it's like run compar and oh okay oh I see and then like I always toggle the only so I don't want to see like the things are similar because yeah llm training conf have like 200 parameters so I just want to see like what's difference between the training

runs I have a few questions some of them are from the Q&A I'm gonna start with one of my own but then I'm G to try and get through some of the ones that are in the the Q&A one of the things that yeah a lot of these graphs are very showing loss the eval loss as a function of something else especially in the

context of conventional machine learning like we always just like loss is a quite good proxy for how good the model is with llm fine-tuning I've never been really confident that loss is a prox for output quality and as a result I spent a lot of time in expecting the output of the model and I haven't formly like looked at the correlation between having a

lower loss and me subjectively think the the the output is better but I have like an intuition that it's not that tight do about this and do you see that people are frequently just trusting the loss is a really great measure of model quality or do people do you see a lot of people are just inspecting visually inspecting the the output samples I think I think you have during the course you have c a lot of the evils

and I think it's really important to build an evil data set and and with generative AI like with this llms you you need to generate samples and yeah ideally create kind of a test that enables you to compare if the model is better or worst and kind of what eval harness does it computes metrics on different benchmarks but that's like for a more generic fine tuning in your own use case you will need to compute your

own Benchmark for your own task but yeah you need to look at all samples you need to put them all in inference and run those benchmarks actually when you run this benchmarks here this project like yeah there's all this hidden sections is because we have the actual samples there so you can go and see what them all answer on the mmu whatever example number five and see what the answer was so you should you could use eval harness

create your own task and put in the format of Evol harness and it will work because we have an integration with Evol harness but yeah I don't know like we have developed some llms not fine-tuning but internally and yeah basically we ask ourselves to create our own data like five question answering by engineer like times 10 you have 50 twice a week and you create easily a data set of a couple

of hundreds of samples that are high quality yeah score your M against that and and during training yeah I suppose the loss is a good proxy of things going wrong but it's not a good proxy of like if you have like five fine tunings you tweak three parameters you will get basically the same loss and maybe one of those is better like it's not easy to know I don't know what's your experience so you asking about

Hamil's experience my experience is I just look at the samples I I think that you know like I said I think it's powerful looking at the the samples and actually Vibe checking that it's correct that's a good proxy of the things getting getting wrong and actually you need to look at the samples because sometimes there's like formatting issues that you are not aware and like you are I don't know there's so much abstraction about like how the token you actually

need to look at the token ID sometimes decode the batch and see if I had a problem couple weeks ago of a mall that was spiking at some point and to actually see what what was going on I decoded the badge where the all spiked and there was like some characters that are not on very out of the distribution of the token I so there was some Korean on an English only data set and the all

just Spike there every time even if I tweak the learning or whatever so go back in time rewind decode that batch see what's you are feeding them all it's easier said than Donna because like if you have Ag gpus and you have distributed something so at some moment like keep a checkpoint reload rewinding time and but this liaries like Transformers act lets you do that like lets you like index on the data loader and see how you are feeding the data to

them all yeah data in this interface because it's too restrictive for me so I end up building my own data viewing tools but yeah I mean that's that's what I do but I agree it's you have to look at these data points somehow yeah yeah yeah and you I think you got you have to look at both like even the token ID sometimes get there's a tokenizer step there that sometimes doesn't do whatever you want like there's like a I

don't know like if you know about the people watching but you have like the tokenizer but he's also batching stuff together so like sequence have now the same length and and now we have more complexity like Axel all do does a lot of magic for you it does multi pack so it put multiple sequence together and like then batch them so you can feed the gpus yeah efficiently so yeah the a lot of things could go wrong there

yeah so decode your batches and check what's what you're feing them all my how did you concretely with a LEL you said like okay you were witnessing a training run you saw I I guess you saw a a loss Spike and loss and then how did you correlate that batch with that loss with this tooling like how did you it's not it's not an easy it's not that easy yeah it's it's not not that complicated but

it's not as easy as I actually you have to patch the trainer the highing face trainer the that runs the training loop on a LEL okay so I I PCH I replaced this the step method and then I I was checking the GR and the loss and whenever it spiked I will decode the batch and log it to a weight onli table so I could like see the step and then see the table and I will save a

checkpoint of them all just before because what I wanted to do there was skipping that batch because okay that batch is bad I don't want to update the weights of them all I want to just compute the loss it's bad I don't update the weights so you skip the optimizer step but like in a lower level Library like you're using like Axel on top of trainer having access to that Loop is complicated it's not as so yeah you need

to take the diary and actually override the part I encourage people to try that kind of things and you could print them on the terminal you could log them the cool thing is that if you log them to something like a table here you will have the run and the step and you'll have the table that has like I put the name on the table like the step where it was logged and the GPU where it was

logged so it was like G V7 I don't know and Sample two was having that Spike on the loss so you could actually log the law separately per item on the batch instead of like Computing the mean and that's something also that could be cool to do of course if you use like a lower lever Library like torch tune or or pure pie torch it would be way easier and I suppose like big companies or big players like mistra Len like log

tons of proxy metrics from all the nodes and all the different stuff they Computing and so they can spot this issues quickly restart and like not lose money basically yeah if you just continuously log all this data whatever is there a storage cost of WS and bases like yeah okay yeah know what that is is it expens yeah I don't know the pricing I

think it's fair value for what you're getting it depends like it scales on usage so if you're using like personal it's going to be cheaper you get like a corporate license with all the security features deployed on your Cloud yeah it's going to be more expensive but there is a cost for the storage yeah yeah yeah yeah there is but you could use like you could use you are not forced to use our storage you could

bring your own like buckets if you put your Google Cloud buckets or your Amazon S3 buckets or aure buckets you can just reference the artifacts to the bucket you are storing the data on and you still get the lineage and like the all the complete TR traceability of what experiment use what data can you tell me a little bit about the syntax here like a lot of people open Wass and bu says they get confused like like okay you're showing that table

right now like the One You're just showing it's kind of like sort of looks like a data frame why why do we even show the syntax here like like that like do you ever edit the syntax or like yeah this this is like a row table so it's like what actually get dumped by the integration here but probably yeah when you want to explore these probably you don't care about all these columns and you want to

yeah see the raw prediction Maybe next to the data so probably you're going to like remove some of what I'm saying is like you know you on the top it says runs. summary and then in yeah so like can you like filter The Columns by changing that or something or what is yeah probably I don't use it like I actually I use the filter here or or like click the columns yeah there's like a specific language to actually query the tables

here but yeah you only need to learn that I think language I just curious okay so yeah I think it's better like using using the the UI that's not that so yeah you're maybe you can Group by here and let's see groups like yeah we automatically try to get relevant stuff here but yeah the group by doesn't make that much sense here and reset the table

yeah I know but here are basically to explore the individual samples of of what we log so as you said it's like a data frame on on the UI so yeah this is mlu had different pieces of Benchmark so yeah we have a bunch of tables here 67 to be exact if you want to dig dig dive deep dive on here it's and probably you should not turn on every single mod because what's happening under the hood is where like

concatenating all the data from all the results and you probably want to explore these tables like one by one for one mle at a time and not all the mes at the same time so yeah I think this is kind of a speedrun of the weight Anis mods features and experiment tracking what everyone most of the people know here so yeah this is kind of the the quick wins I show you like you can save your

workspace view so you can organize your data save it put it a name like this is my Evol view this is my training View you have the run compar that quickly you can see what's different you can pin columns to bring them to the left mostly if you are like tweaking learning R batch size probably you want to Ping those so they're always visible and there's more charts like you can explore there's like the parallel cordian fla I

think there is one here I quickly show with the beautiful lines that lets you like explore if you're doing like hyper parameter optimization is very useful so yeah we can say that reproducibility in machine learning is is key it requires some effort but there are quite benefits of of using it to like this and and having reproducibility beltin yeah when for me at least I don't know means that your screen is shifted left or something like it's not

within the view or is it just me Dan can you see his whole slides yeah I think it's just you it looks normal to me okay sorry okay oh so yeah so no for me yeah this like a standard point of reproducibility but what I mean is for me the most important one is like I come come back six months later I and protect against myself and and actually be able to to reinspect my project and maybe continue

working on that or or handling my project to someone else so they can keep keep adding like more experiments or understanding what I was doing so yeah this is like a nice illustration from Christy weeder I encourage to check it is it's yeah it's a presentation that's online you can check it out so it's step by step like using yeah versioning your code would be a step for reproducibility maybe next step would be like putting everything into a container maybe using

tools like weight and biases to keep track of your experiments organizing this like when you start collaborating with people so there's like multiple steps and like there's no zero and 100% it's like you can you can be in between depending on how much complexity you want to add to your project so yeah that's that's the take the take out here so there are different tools and Brace tooling that helps you accelerate your experimentation and come back in time and be able to

rerun your experiments so just to quickly put perspective here what I show Mostly is what we call now models it's what you know from weights and biases the experiment tracking part so I want to talk a little bit about like we have a new product that it's it's tailored for like okay you have fine tun them mall now and you now you want to use that mall so you want to put them all into usage internally or externally and you

want to keep track of of everything that's getting in and out of that all so we are trying we're excited to like show you here our new public preview project protocol weave that yeah serve that kind of persona that's it's playing with the Ms and needs actually tracing and evaluation so as there as H was saying like this table sometimes like very clumpsy when you are like running and you want to see inputs and outputs side by side and yeah and you

are using a lot of mods on API like you're not always fine tuning so this new tool serve this kind of persona so before diving into this I'm going to just tell you that yeah this is another python package so it's not nothing fancy pep install weave and you get the package installed it's a lightweight Library that's built on top of pante is is very easy to use and it's actually when you start using mson API you will

probably have a code around your API call it's not going to be like call openi and get me the raw result so you're probably going to have some python code to pre-process forages what do you mean it's on top of pantic like can you tell me what that I'm gonna you will you will get it in a couple of minutes okay so we leverage pantic so this this tool enables you to keep track of the Python functions

you will you already buil to make those calls externally so you just need to add a decorator so import weave we. op and we will keep track of that function so yeah we use this decorator to read the signature function and yeah and see what the mod expects as an input here and what's the output and we also have Integrations with the popular vendors like openi so we also keep track of the underlying opening ey call that gets

called inside this function so before just diving into the library I'm I'm going to just propose you to to that we try to play with with a words problem that we may try to solve with an llm i I had a lot of fun playing with this a couple of weeks ago and I I thought okay world was fun but the New York Times has multiple problems and I don't know if you have played this this

we called connections so it's like 16 words and you have to find four groups of four this looks like a problem tailor for an llm or at least that an llm should be capable of doing we could maybe add this as an eval to whatever eval harness or something but it's not as easy as it looks like so yeah these four words form a group there's an unique solution so there's four groups of four words and you have to think that

maybe if you put that word on that group you will not be able to compete the next group because that word also belongs to other group so you have to find relationships between words that's kind of an easier relationship but yeah and then the problem actually when when you fail one of those words it gives you like three mistakes and it tells you like you're one away so one of these four words even if they look like cow

related words doesn't belong to the group and yeah that's the F the final solution to this problem this was like one week ago and as you can see like the last purple one is pretty hard like it's bull flea meat stock there those are markets and it's not as easy like you need the to think out inside of the box and maybe a person that is well versed in English I'm not a native English speaker so I have a hard time with these

problems but I have seen some people that are really really good and can can actually solve this 100% of the time I can't I always use my three trials and most of the time I failed so I'm also pretty imp impatient so we try and like get something that works so yeah how would you solve this with I know opening igbt 4 you may create a problem that explains the puzzle and yeah and maybe give the list of words and and just ask

them all like hey provide me a solution for this maybe you are better prompting maybe you put like some examples like hey a connection types of fish is a connection of bus flounder salmon and trout and maybe you give him some hands like few shots of harder examples like that fire things that start with fire but may not that may not help for that specific like 16 list of words you gave here so if we try that prompt and it

gets you something like this so it gets one group correct the anagrams that's not as easy but I was pretty impressed that this is a good starting solution actually and the others are like three out of four and I mark on red the word that doesn't belong to that group so you could like iteratively solve the problem like you could try one prompt maybe like Point him out like hey this group have three out of four think

outside of the box maybe use at tourus to actually call and get the meanings of the words but like actually what I wanted to solve with an llm here was the hard version of the problem and the hard version is give me the full solution at once not not giving me hints of like hey I'm three out of four and as someone pointed out like a a good person that speaks good English has good literature knowledge should be able to yeah find

the solution at once maybe on a piece of paper you write multiple combinations but you you find the solution at once so we can use the LM as a scratch pad that means that we can call the LM multiple times and maybe create intermediate Solutions and then use another llm or another python code because most of the time when you build apps you have like C yeah functions or external connections maybe a rag actually you could try embedding the

words and searching on embedding space and maybe finding a relationship of the words on that in space and maybe that proposing a solution I'm curious if if that that works actually and then finally just submit like so you can run whatever you want but you need to submit the solution at once and it will be good or bad like wrong or correct so the idea here was trying to solve the problem together collab was very finy a couple minutes ago that we

can try anyway but so we how do you solve a problem like this I think Don and haml have show this extensively you need to create a evaluation data set so evaluation and then you run your Mall against that evaluation you can put a score in this case like how many words correct how many groups are correct on the 16 4 3 1 four is perfect three is almost and yeah zero is none and we put

together this dashboard that we can you can log and and we can see if someone finds a good solution you not it's an open project you can you can see and inspect what people are doing so I wanted to show you this so I build a collab so if that's the link 1 db. me/ connection and it's a very simple collab so it's it's going to try to solve this as a very naive approach so basically the

prompt I show you on the slide and and yeah it requires only we an open ey so if you have your open ey creds that's great you can use whatever M you have available so as I tell as I said like ideally don't don't take here that will log the results to the public project project so we can have your results so maybe iterate a little bit play with this on your own like account so you can mess and maybe create

a better prompt maybe iterate multiple prompts and then at the end when you already have a good solution maybe you chuggle that on and you push the evals results to the Le board so we can inspect what you're doing and basically to use weave our project to trace our product to Trad you need to import weave this connects to weight and biases and tells him okay you need to log all my function calls to that project so we

are logging to the connections project the first time you use weights and bias you will be required to paste your API key so yeah basically you click that link probably if you have use weights and biases yeah I have SSO so it asked me for that every time now forgets about this you know already pass time so just let you know in case okay you need to know yeah so yeah key let me download a data set here

you can run those on your time so yeah the data set is like the list of words with the actual groups you have an openi key so we use that so I just show you before this is what you will do like you will have a Cod openi and you will put that decorator on top of that and that will be able to if we call this function you get a link as you always get with

weights and biases you get a link to the actual code so it error out because yeah this is a nice feature of openi that you can expect like a Json format output so I'm asking for the capul France in Json format here it will fix that we will use the Json output afterwards so what what do you get with that link when you click that link if you click that link here you redirected to the weights and biases with

dashboard so I have played with this project extensively so you see that Coop the function we defined here all underlying calls like the actual openi chat completion so we know that yeah we can expand here and see okay we call with that list of messages we call gbd4 and the max tokens parameters we pass and we got an output structure on on Json format here so ideally what we want to do here is actually solve the problem so we we

are going to just Define a function to to par the Json we recover because the Json looks nice and we can actually put everything together in something we call a mole that's it's light wrapper around a ptic base mole so it's just to keep everything organized and it expects you to Define just a predict function that knows like how to compute how to call them all so this is the oneshot m and we're going to use the same pront I

show you here and if we call them all on those words M predict here we will get also a trace to that specific and we it produces a solution so if we click here we get the the model output on the predict function so this function called the the generate solution and that function under line call the so this is like a debugger so you have like the traces of all the functions you are decorating and how the functions get

nested so when one of the functions fails you can see what the inputs and outputs where and click here to expand everything and so we have the promt the com and everything gets version so so we have the one shot M here that got version so V the F version we have created of the oneshot mod so as I said before probably you need to create like data set in this case we have a data set and you need to

create a function to check the solution so here I'm just checking that the groups match so check the solution this solution is perfect so the all this is an easy an easy not every day is as easy but this one was easy and the solution was correct we can click also on the link and we see that the check function yeah got a perfect match and we can expand and maybe compare manually if the match is

correct and to actually submit here H an evaluation you can run this so this a wrapper that basically says you are going to run the data set against them all so we're going to run them all against that data set and we're going to score with the scoring function we Define yeah this is an a same function because we run all the calls in parallel and here we got like evaluated 20 examples and we yeah we aage the

scoring function in this case we got like three correct out of 20 not that great so if you can beat that it would be really cool and if you click on that link you are on an Evol valuation dashboard and this is the the original dashboard I show you so we have like a small leather board here and we see the actual true fraction here 15% I got only three out of 20 and I can have versions

of my evaluations of the mods I have been using as I as you know as you see here have been playing with this for a while I have multiple mods and yeah I encourage you to try it and maybe submit something here if you want we can we can dive here and see how the all as you were asking haml about how the all performance on every sample so we see like every row here is a sample so here

the match is incorrect we can dive here so the one should M failed and they generate solution we can inspect the solution and compare against the groups it generated and why the check solution was wrong it got just two out of four so it's very handy when you start creating stuff that gets more and more complicated and called external functions or maybe even external apis so I want to just finish with a slide give it a try the collab should run

and if you have any questions just ping me on Discord I think I ran out of time so that was pretty Speedy so if you have questions or we have us time for couple of questions aush has been answering all the Q&A pretty pretty well I'll say since we're out of time let's head over to the Discord and kind of pre it there yeah there's there's some questions about getting code but those

the sort of thing I think it'd be nice not to answer on the spot where it'll get lost anything you write in the Discord people will have access to for a long time okay cool thank you guys all right thanks so much see you ciao bye