Lecture - 1
And it makes me wonder. All right, stop it.
Hi, everybody. Welcome back. Our ton of things to tell you.
So many that have to make a little, little case in my head. One moment.
Seriously? Give me a second. Okay.
Um. Okay. So there are more.
Well, um. And then.
Got it. Hey, the class starts now.
So, first of all, happy Ramadan. Okay.
If you are Muslim celebrating Ramadan or if you're Muslim friendly, wishing Ramadan happy Ramadan.
And in the same breath, Happy Nowruz. Happy Nowruz.
Cool. So it starts tonight, right? Perfect. So wonderful.
There's always cause and time for celebration. Right? And there's also a cost in time for not even removing a comma.
Okay. When you are at the very top. Right. Don't touch it. Okay.
That's search engine optimization for you. But wait till I show you what Google has planned for you.
Okay? Google is getting more and more strict about people that actually basically, you know,
muck with their rankings and want to go to the top their anti-spam policies.
They have some pretty strict updates this year. Okay. I'm going to show you a page pretty soon.
But before that, I wanted to tell you, I hope you didn't forget everything.
Right, because it's been a week. And so everybody was off.
And I hope you took my advice seriously and just basically goofed off. Just blew it away.
A week went by and you did nothing. Which is actually the best thing you could ever do.
Okay. Yeah. Sometimes I'm in the middle of doing I'm busy doing nothing and people bug me.
Okay? And I'm looking okay. Leave me alone. I'm busy doing nothing.
So you should also be busy doing nothing. It's actually. It's. It's. It's. It's a thing.
So, welcome back. We only have six more weeks left, so nine weeks are done right?
That is like a 3/5 over. That's our term is always like a PI.
And the pi looks very interesting. It looks like that. It's like that, right?
We do that. Actually, that's one, two, three.
You know, forget I did that. Okay. So more and more like.
More like that. Okay. Uneven. But what the [INAUDIBLE] you got I.D.?
Okay. Three, six, 60 divided by five.
Okay, so, you know, guess what? We are done with all of this.
Nine weeks are done. Ten, 11, 12, 13, 14, 15.
We all done about. As you know, the break is not done. The middle.
The midterms course came out. Oh. So what do people think?
You can be honest and give me your reaction, your feedback, your concerns.
You're dissing me here. You can show it around. Mr.
Anything to say? Making the new What?
Oh, yeah. Okay. Yeah.
Like, some of the questions were like, what?
The will. Specific answers when there could be more.
So true. So, like, I'm wondering, like there's any loss of.
Yeah, that's actually maybe they're the number one.
The the only thing that I wanted to bring up, you know, So an exam is like, you're looking in the mirror, okay?
With their flaws and your words and your pretty face and smile and everything. Right?
It basically tells you the truth. In other words, you might have studied for the exam in your own way,
and you obviously know a bunch of things, but the exam asks you something different.
No, I mean, it's something that happens in life. Interview questions go that way.
Okay, Your life goes that way. So sadly, you cannot always expect to ace everything and get 100%, you know?
So in this exam, if you didn't get full points for any reason, don't be like too disappointed.
Just tell us off. Wow. There's one more way to learn this. And I didn't know.
Okay, But that is not the same as a somewhat like an ambiguous question.

Okay, so my questions are very open ended and I want them that way.
So when I interview people, that's the kind of question I ask actually, potential people that I might work with because I want to see what they do,
like how they how they respond, even if they don't get the right answer.
Okay. I like the approach. They call it the approach. So the one with the Embarcadero Freeway sign is a classic example.
Okay, so maybe the best answer would be a skip list, because that is actually what it is.
So the express lanes, right? There are fewer exits.
So in a link list, you don't have to jump from link to link to link, you can set up extra skipping links and go back and search if you want,
but if you don't go back, you can keep going forward and very quickly eliminate something like not being in your list.
Okay. Or you can quickly narrow down to something that might actually be in your list, but there might be other data structures.
Also possible a tree would be one of them, you know, but I think a covered tree a little bit.
But even if I didn't, you might know Tree by yourself. By a tree.
It's called a tree. If you're not there, you can file for a regrade and you can expect definitely some answers to some marks.
Okay. So I'll go over the rubrics with the grading people for the regrading purposes,
so that will relax some other things and basically allow for multiple interpretations.
So if you did that, definitely please file for regrade and you will get some points, you know, but regret is obviously a very structured thing.
You know, I've done it for like a long time and these things work, by the way,
in the past, ten years ago, 20 years ago, even though it's zero, I think already great.
Okay. It's one of those things where you get your answers and watermarks you got your stuck with it.
But we don't do that. We actually give you a possibility, a chance to go over and maybe potentially ask for more points.
So this would be an example if you thought that the question was too open and the rubric said something and you wrote something else on your deserve,
that your answer is still you feel that your answer is correct, then please fail.
Okay, So you cannot fail. Regret, of course,
for no reason asking is not free because otherwise 100% of the people that did
not get 100% full points will simply ask for regular just to see what happens.
Right. And that's not fair. You know, you can like waste people's time, so please only do it if you're sure.
That's right. There's a little penalty. The penalty says once you file for regrade will carefully go over what we give you.
And if we give you too much, we'll actually take it away. Meaning the newer points are good, potentially might be lower than what you initially had.
That is simply to discourage you from making spurious requests at all.
Okay, So if you're genuine about what you think you want, definitely ask for like more points.
Because the goal was not to make the exam hard. I promise. Okay. It's never the goal again.
But I also had to do something a little bit different compared to the past two years, because the last two years everything was online.
Everything was entirely open. In fact, the last time I taught this class, I ran said, You can use Chargeability.
Just going ask. You know, that's typical of what it gives you. But I also said at the same time, dawn verbatim copy and paste what it says, you know,
because then there's no then you're not even looking at it and it has hallucinations in it.
So people use it for good advantage. Then they don't have to remember anything.
But this time you had a cheat sheet, but hopefully was not needed as well.
So basically it goes back and forth, back and forth. All right. But then the goal was never to make it difficult.
So the final make it a little bit easier, I promise. What are the different means?
Okay, I'll make it. Maybe fewer questions, more time. General pressed for time.
Okay. And the second half was more lightweight. Not that the first half was pretty heavy.
You see, there's only just three or four core equations we keep going back to.
So this whole field, it can be extremely mathematical. But then that is all abstract theory anyway, about data access and relations and things.

Then it's not practical. Excuse me. So I like to keep things more practical, especially as things change so much.
That's why I always keep harping on what you can do with all this. And I'm going to show you our search engine as an example, GrubHub search engine.
Okay.
So in your search for food, how does GrubHub like, make it so easy to search Indian food in a California kitchen, harness it, show up so quickly?
I want to show you. Okay. Any other comments?
You just walked in here. Any questions on the midterm? Any comments on the rubric?
Yeah. So that's one of the ambiguous ones set in question four.
In question five, you can consider asking for like more points if you think that what you wrote also answers the question.
Okay. Anything else?
White people came in. Yeah.
All right. So we can actually do a little bit of the extra part, Right.
So checked on two little things. Okay. Or maybe three, including this one.
All right. There's something very interesting that is happening. In the world of software engineering.
What it is, is it's almost like, you know, when you go back to 1950s and sixties,
people had a big computer and they had mechanical switches really once and zero, almost like you see on the wall.
And people sat there with a piece of paper where they had written the program out by hand and
mechanically flipped the switches on and off and turned the entire machine on and they got results.
Okay. That is basically what computation was all about.
You had to actually write machine learning, machine language code, you know, then when Fortran and alcohol were invented,
we call it a higher level language where suddenly you have things like five from 1 to 10.
So you can actually understand what that means. I'm trying to look something. Okay, that was considered a very big breakthrough.
Then we never have to go back to the assembler language with things like Move.
Register one, come registered. Two. Hello. Register five. No.
Or store pending communication. We don't have to do that anymore.
Or we don't have to program in ones and zeros anymore. But why would Java or JavaScript or Python with all the amazing syntax we're so used to now?
Why is that the final destination? Like what? Why can it not be also like the of like, why can't it not become obsolete?
I'll show you somebody so more amazing. No less than Jensen Huang mentioned that in one minute.
Okay. But this one is from a business called Professor, right?
Undergraduate Education Dean. He talks about a prompt engineering.
It's like a new job role. Okay, But not really, though, because it'll go away because, you know, and alums can optimize their own problems.
Okay. So you're you're not going to have a 20 year long career becoming a product engineer temporarily.
You actually would have that role and all that. That means, believe it or not, you have a little search box, so that's your competition.
But if you walk up to it and bought a real order type, whatever you want, so you do your thing, you know, she does our thing, bought it enter.
You get high quality, amazing results back and then she gets almost nothing.
So you walk away with the job. It's something in your head that is typing a search engine.
Okay, that's called a prompt. So prompt engineering simply is like a hacking form of hacking,
which says I have mastered what I should put in and it's my secret weapon is my supervisor.
I won't tell you what what I'm typing, but I can get you better answers than the competition.
That is called prompt engineering. But then it's crazy, you know, you can't really learn that.
I mean, you can, but it's basically very specific.
So what works for Gemini might not work for, you know, Jupiter 4.5, for instance.
So that's a pretty silly thing, right? Plus the alums can themselves, you know, generate better problems.
Okay. That's what's really happening. But what is also what can be done, there is something very cool.
It's basically what you would call, you know, Asian programing, Asian programing.
So Asian programing is where you want a machine to do something pretty complicated.

Last week during the break, some of you might have known this. There was an Asian that was released called Devin.
So Devin does many things that a software developer would do all automatically.
Okay, completely unguided.
It looks very scary because then the the news media that are hybrid would tell you while there going a software engineering role you know so
not so fast okay it's not going to disappear that fast but what I mean this is actually something very cool I'll show you that in a minute.
But Devin does task after task after task in some specific order that it comes up with, you know, the system comes up with.
But that might not always be the best order, but there's no need to stop here.
You can program that little chain yourself. Okay. So each piece is called Start.
Each piece is something that you can tell them to do by tapping a prompt,
but this a series of prompts that generates automatically and then adjust them all one at a time.
But if you don't like that, you can program your own. So that is a very cold and that is not prompt engineering.
It's more like architecting sort of the result becomes like a piece of software.
So much like you can have a phone loop here if you wanted.
Okay, but you are not using Java syntax and not using python syntax using Asian syntax in things like Lama for example,
Lama has something called my index, you know, or something called lang chain.
So Lamar Index lang chain.
Those are the new languages to learn because using those languages you can make prompt structures like this, you can even branch.
Then you call it chain of thought. So there's basically tree of thought, chain of thought.
You know, it's collision programing, It's called many things, but it's a very powerful idea.
Here's why it's powerful, because the more you let,
the more you trust the system to come up with something long and big like that and the more it's going to screw up.
But here what is happening is we know what we want.
Okay, in a search engine, I know that the initial thing that I have to do is go to the Web and actually get documents together.
And next thing I want to know is I want to generate an inner index.
Next thing I want to do is some kind of ranking algorithm. I know all of the steps.
And then I tell the system to write code to do all of the steps, and then magically I have a search engine.
But then there's not a single energy programing in there that is useful.
Okay, that's like Asian programing. So I'll tell you about that. I show you who said something about soon, but meanwhile, this is a little bit weird.
Yeah. And it's also true in the meanwhile that, yes, things like Java Python will become obsolete.
Exactly. For this reason, because this can write a piece of Java code that could create and read an index like your homework.
You know, just imagine that with enough training, with enough data,
we can take all of your homework solutions as data and then train a system to generate or, you know, and aggregate.
Okay, That's basically how it works. So it is possible. Okay.
But this is not my main thing that I want to emphasize, but it is true in in a broad sense that spawns a new generation of governance.
The US economist yesterday said there's no doubt that he's going to trash a whole bunch of job categories.
Actually said that again. But if you read the whole if you watch the whole interview, [INAUDIBLE] also tell you it maintains spawn even more roles.
It's one of those things where the graph to an economist looks like this.
Okay, summer time number of jobs lost.
Okay, so currently the US has so many jobs at the end of the year.
What he's saying is this okay, and this is bad news for you guys only.
So for a while there's going to be this freefall and we have no idea how low it's going to go.
But in the long run, it'll pick up even better than where it was. That's basically what he's saying.
You know, this is about GDP. Okay, Imagine that GDP. So he's saying in the short run, you know, jobs will actually go

away.
We should say that. So, yeah, this all is very, very interesting.
You know, mixed like news that you're getting. But mixed news does not mean contradiction.
It could be both could be like bad for the short term. Watch this.
I'm going to say jobs. Yeah.
Air will destroy employment in some areas. Okay, let's see what he says.
Today. By the way, I don't want to go back to all topics I want. Hey there, Brenda, it's Carol.
Exactly. So which are we operating on? You mean arm?
It's all connected. Asking the right question in great detail.
You sure you're an orthopedist? Actually, I'm a Sagittarius, especially when it comes to your finances.
Do you have a question? Are you a certified financial planner?
Yes, I'm a CFP professional. CFP professionals are concerned.
Marketing is an altruist. That's why it's got to be a CFP.
Find your CFP professional. Let's make a plan call. Okay, So let's not today.
Do you see I as a job creator or a job killer?
Well, I see it as a productivity enhancer.
It will destroy employment in some areas.
I mean, there will be parts of the labor market in Germany that can be replaced to a to a degree.
But then you'll also find other ways of innovating and and creating more jobs somewhere else.
I mean, this is the world right now.
I have no idea what that to this innovation for you guys can think of as of year is that you have an innovation that is basically
labor saving and that reduces employment in some areas but then boosts that and in others hold that balances electricity,
water, I think different printing press, but on and on and on.
How much more confident is that? It can signify programing languages to growth over over time by basically boosting productivity growth.
And we have actually lifted our long term growth estimate somewhat.
In the long term, the impact of air to air, it's actually made you more bullish on the US economy from a GDP growth perspective.
Yes, that's correct. I think the biggest impact probably won't be felt for another five years.
Also, all our upward revisions to growth have been sort of concentrated late this decade, early next decade.
But we do view that as an economic positive from a from a GDP growth perspective.
Yeah. Asked to say it gets worse before it gets better, you know, and you are in the in a slump.
Okay, now try it out. Okay. So this is very true, though.
You know, we're all stuck learning brand new things now, like a Lamar index and a long luncheon and so on.
And so it's going to be okay. The next thing I want to show you is actually don't tell the whole other jobs.
You know, jobs being distracting. I'm going to say 2020 for GTC as we speak in San Jose.
This a biotechnology conference going on with Jensen Horn.
He's basically the god of Nvidia, right? Yeah.
So Nvidia has a blog, so I'm going to say and read your blog.
Let's see, we'll just summarize it for you. Okay? The conference started yesterday.
He talked amazingly about many different things.
He talked about a processor called Blackwell. So Black was an incredible AI first chip.
Many data centers are going to use that, you know. And so then let me I'll tell you what he's going to talk about when it comes to programing.
According to look at this, you know, this guy is iconic and gets better every single year.
All right. Cecil Blackwell computing platform. Okay.
He talks about microservices first. Okay. This one, we need another way of doing computing.
Actually, partly. On the flipside, it means another way of creating software.
Okay. And there's more coming up on that. All right. So entrepreneur's mastering.
Okay. Blackwell platform on this real time generator, 8 trillion primary language models.
Okay. I mean, Jeopardy 4.5 supposedly has trillion parameters, but that's going to become the norm because our

process is like Blackwell.
This is cool name Nvidia inference microservices.
In fact, if you read more, they actually call it a containerized microservice.
In my database class, I use an acronym called MSC. Makes it like this.
How all the future software should be done in my opinion. Okay. And slowly we are getting there.
M stands for microservices. C stands for containers like a Docker container, you know, partition all the containers.
And then the other C stands for cloud, such as a GPO cloud that is going to talk about.
So when you write code,
which you call a microservices like a function call and then you put them in containers so we can have many instances and scale up
and down based on need and put them on a cloud so that you don't have to have local computing beefy little laptop on your laptop.
Then magic happens because each one of us can connect to many millions of CPUs and then get a work done.
Okay. That is what he's talking about. Okay. See, that connects developers with hundreds of millions of GPUs.
Jensen Huang does not make a [INAUDIBLE], okay? And you can go back to all this.
Since 1993, everything that is ever said has actually come true.
I'm very skeptical about many, many, many things. I think most of the people are like B.S., hype, okay.
Including Microsoft, including Google, including Metta, but not on video games.
I always had respect for one company and this and video. So I believe, like what he's saying and I think it's going to come true.
It's cool the way that's going to be done this again, with this whole new platform,
because if you are able to simply pick from these bunch of function calls written in whatever programing language and put together their application,
even applications like these, then that is crazy. Say this is written in Python and Ruby was written in Java.
It all doesn't matter to us. We're just simply wired them together. Okay.
That is the whole idea. If you just google it and really anime will actually see an image is great.
And then he also talks about Omniverse, which is my computer graphics thing.
You can make you can make renders that look exactly like this room 100% for the real unless they tell you it's a render, nobody would know.
It looks like a photograph. Okay. You could do that in real time. So this happened yesterday.
So please watch all of it. 2 hours. It's like gold. And then they have more things.
They have 1000 sessions, a GTC, and they'll all be on YouTube, thankfully, and more than two thirds.
I. By the way, the other one third is computer graphics. Okay. In the past they were all 100% computer graphics, but there I no the biggest customer.
Okay. So I like all this in a graphic and it all does a great cool.
That's data art, by the way. Okay. Okay. So look, he means he's going to talk about larger models and on graphs.
Black culture. Yeah. So this black wall chip that you can hold in your hand replaces entire buildings or maybe entire processor that weighs 3000 tons,
which is like a heavy SUV. So imagine isolating a whole series worth of computing with something that you can hold in your hand.
That is actually what already happened. Okay. So Blackwell says all of this right got.
So wildly cool. And then future generator in our scale up.
Right. Okay. One giant zippo, this couple. Okay, so if you have this one, see that?
Datacentres are like air factories. And in it's all about data. Look at all the people that are excited by Blackwell.
It's basically everybody that you've become household names in tech, right?
Everybody. So search providers, obviously. Okay.
So for sure, it's good they're going to use it. Everybody. Okay.
Just pay attention to this one. Rather than writing software, you assemble a models by a model image agents.
Okay? You write them and then you assemble those individual agents and they write the code for you.
Okay. So then you can use names to actually program them. Cinema might become an alternative to things like le my index la my index.
And then also this thing called long chain. Long chain came first in LA mind extreme afterwards.
Okay, the bot do the same thing, but it might be a much, much more worthy, you know, more, I guess, more flexible

replacement.
Look, this is in either one of them. You write that see that land chain index?
You can go home and compare like all of these to that interface and you'll see right.
This change of tasks and then there goes and solves it for you and does tasks might even include code generation.
Okay, even documentation of coding basically anything at all.
So look nuts. Yeah. So some of the agents they talk about are actually programing agents.
In other words, you have things like a GitHub copilot.
So now it talks about a copilots plural. Actually, there's a there's a piece called.
GTC. Hmm. Now, let's call the Deputy Dart CEO, I think.
I'm not sure. No.
Not done. Hmm. Okay. That's basically an Asian.
There's an Asian market place. It's brand new. I'll find a link to put it up.
Okay. So these things are evolving so quickly. Tangibly?
Has a store all right to open air, Has to talk outside. But that's not what this is.
In the Chargeability store, you can make a trip to, like, a Chad Asian and sell it.
But this Asian marketplace, you can make these kinds of Asian, you can program them and you can sell them.
Then imagine the new you as a programmer.
You can just go shopping for alterations and acquire them and make search engines and like dandelions or something, right?
I mean, it's ridiculously powerful. Okay, so I know it's not Klaviyo and I probably won't find it.
Okay. It's not. This is something Dark SEO, you know.
Hmm. I'm not sure if I'll find it. And then I should have domain call and ask you.
It's okay. Okay? It's not any of this. All right.
So things are changing rapidly, so pay attention to what he's saying.
All of this will all become too hot this year. The last thing I want to tell you is before we get to our stuff.
Maybe GrubHub. GrubHub.
Such, actually. Sequester that I think I mentioned probably a couple of times.
It's amazing software engineering site. So please go to questor and sign up for free for the newsletter.
You will learn so much about massive systems engineering.
You know, how do you do microservices, How do you do the front end in order to in order order.
Okay, So I'm going to go to the blog and show you something that is Brian. You just put your email in there.
That's all you need. See, look like so cool. All of this, all of this.
Look at microservices DoorDash. Yeah. Uh, okay.
DoorDash search engine, not GrubHub. DoorDash search engine.
This is what I wanted to show you. It's going to remind you so much of your homework because the steps are the same.
Okay, so how come you can grab your DoorDash application and start searching for any kind of 41 instantly lists like Nero's restaurants.
All right. They have to do an inverted index. Okay, so here it is.
See? Look at that. They use Elasticsearch and then Apache LUCENE.
They go together. Okay? They look open source.
So then they make search engine programing a little bit easier and then see the document indexing, Right?
Exactly. Indexing. Writing code for. And then they could increase the whole thing.
Multitenant just means, you know, we can live on like so many different clouds, like parts of all of these.
So that is what I'm going to show you. And then every issue of the Questor blog has so many extra things.
You need a caching, for example, just learn, learn and okay, they're going to teach you.
Okay, so what about DoorDash Search engine select right here.
Right. 40 million users, hundreds of thousands of restaurants.
Those restaurant menus. Correct. Are there documents?
So that is what they ingest and they look for keywords in them and find an invalid index.
So if I type Indian food, what restaurant should they name?
And also how many miles to closer to you? And they can rank demonstrate.
Okay. So this is very neat. Like this search which only returns stores, avocado toast, you know, then hipster joints close to you.

Okay, so here scaling then that's how they make it pretty big, right?
Meaning you can do. Okay. This section, what's super in your homework is very small.
Your data. I'll try to give you a like a bigger version of data, smaller version of data.
But even that sometimes replica cannot run anything. How do they serve 40 million people, right?
You need a way to scale everything up. One way to scale things is exactly.
I'm Sisi, by the way. You can make your search engine run as a microservice across many different cloud missions.
Okay, so I'm not going to one simple you, but you can also use this product called Elasticsearch, and that is where the elasticity comes in.
Again, we're able to scale the resources up or down based on need.
Okay. So where you have select this document index, you take the document index that you make, right,
split the index up and stored them in many different machines and search through all of them.
So that way you can get centralization. Very wonderful. And then each piece, when you break it in many pieces, each piece is called a chart.
It could take a piece of pottery to break it up chart.
But you cannot just only one shard, because whatever the shard is in the machine and the machine feels like you need a copy of the chart here as well.
This chart on this chart of the same piece, these are the same piece is the same piece.
Those pieces are called replicas. So you take a table. An index is like a table, right?
Now, first thing you do is you break it. I take a table like this. The first thing you do is you make these so-called charts.
That's a charade. There's also a chart Chart chart called Horizontally Fragmenting It.
You're going to call them chart one, two, three, four.
So my big index with the letter A's here on the documents for a zebra was here on the document for Z Classical in my index.
I break the index into four different parts and store them on four different machines.
And I also make copies. So this call chart one, I make a copy call chart one B and start to be those copies are called replicas.
That's what this talks about. Okay, so you got a chart and also make replicas of the chart, obviously.
Well, but if you do that, then it was actually slowing it down.
Right? That is classically, by the way, how we scale up. But then they ran into a problem with that as well.
So they built their own search engine and then they customize the indexing.
So I have not. Right. Exactly what that is. That's an amazing post, by the way, the blog, the engineering blog that they have Doordarshan doing.
So I need to go back and very carefully like, do what this is. Okay.
So I'm not going to tell you now, but there's something in here that made it be better than classic charting and replicating it.
So it's very cool. I mean, it's wild that they went public publicly documenting all this, right?
They can keep it to themselves. Or even patented, actually. So good for us.
We can then learn from all that. Okay, so we go back here. LUCENE solution is able to index search.
Maybe some of you might have used LUCENE If you go to Apache dot org,
you can then download all of the and then it comes with like an engine controller as well as holler solutions.
Okay, look at this. So DoorDash, again, you know, the index documents in this case, it's all the restaurant menus.
Okay. And for each menu item, the name of the restaurant, that's how the invert it.
And then the former editor and their families read them. Then you have to search and then you have to rank it.
Okay, See this? So these are documents, literally your homework, right?
So index all these words, but live artist upwards so that a word like Breez would be in first document or like Bright would be in the first document.
And also in the third document you can see to literally the index like your homework.
But now these are restaurant food items and then these are restaurant names and even sorted by again,
how close it is to you or how much Yelp rating, you know, what do we want? Price range one start to a status quo.
Then you have the search tenant isolation. So this all about cloud stuff, okay.
Yeah. So then that that's basically it.
So it's very cool when you read this plus this, you will know a [INAUDIBLE] of a lot about exactly how it is.
Again, text fields, isolation, search, tax evasion.

And this is quite new. The search engine modification is quite new.
It means you. And until about last year, they were fine with Elasticsearch, but suddenly they have a need to change.
And so now they changed it. Okay. I think that was the last piece that I wanted to tell you about.
Next is, okay, bringing up the 2024 mark.
Okay. So today, here's why we should do. Roughly spend an hour each 540 right?
647 4820 roughly an hour on these three new topics.
I'm going to do something very interesting, which is basically, you know, for an hour layoff search engine advertising, because the whole market, the way search engine advertising is done is going to radically change, you know, anyway.
So rather than me go through, like all I wanted to ask any of this index exam, by the way, but at the end,
I might find the video from last last time lecture and give it to you so I can at least watch it.
You're not missing anything. I'm here so that way we can do some new things.
Of all the things I have to tell you, this one is not, relatively speaking, not that valuable.
I mean, it's valuable for Google, obviously. But then to know, like, how it works, not to much, but then to know how it works.
These are a lot more valuable, especially the first two. Especially the first one.
I didn't tell you here, Right? I said Asian programing is going to be like one big job category.
I think we need to learn how to make them write code for you.
Second, I think that the whole idea of generating air, but for this you need a ton of content knowledge, okay?
You need to be able to generate everything from business plans to legal briefs.
You can file in court, you know, order a sentence. You got to handle a prisoner like all kinds of crazy things you need to come up with,
right it all under the rubric of generator and then to go along with it or maybe independent from it.
Even this notion of retrieval augmentation.
I've said this again and again and again.
It's probably the most valuable thing about all of the whole alum business, because this is how we can stop the 100% from hallucinating,
meaning making up our fake answers that have actually completely bogus that the grammar looks right.
But then, you know, for example, Canadian Airlines, you know, sold like a seat a few weeks ago, I think for $1 or something.
And that is a kind of bad thing that Jennifer would come up with because it has no idea what it's doing.
Okay. So to stop it, you have to go to external knowledge sources.
You have to go to an actual database. You know, you can even do a computation if you want, but it's everything.
But ask the alum directly. Answer. So one of those ways in which it can be done.
If one of the external augmentations is a knowledge graph, it's a knowledge graph.
It's an old idea. I used to use knowledge graphs in the eighties.
Oh my God. Okay. Like, is that old look at that time it was called Expert Systems.
Have a lot to say about all this. It was always nice.
I always liked it because I knew that it always works, but time passes by.
So during the nineties and I was very difficult to scale up, this went away for a while.
Then when the 2000 started MLA for GP use and cloud computing, all that became like the norm, right?
So people said, Oh my God,
supervised machine learning a label the [INAUDIBLE] out of data in the world and basically throw away knowledge graphs and make fun of them.
Because actually what happened this also called symbolic reasoning, okay, this is called knowledge based A.I.
This has so many names is old. Some people call it go fight good old fashioned A.I.
So I started okay, people made fun of it. But now to me, it's pretty funny.
Things have come full circle because now the air sucks. It makes up like, all kinds of crap.
So how do you fix it from sucking? Go right back to knowledge graphs.
Yeah, but it's good though. In the meanwhile, between 85 and now, what happened was something useful.
Back then you had to learn a programing language like Prolog or Lisp, you know,
or maybe some of you are that somebody wrote using Prolog and left to click on, but you're limited to what you can do.
You cannot just freely like talk to it, so to speak, game. But now with Al-Alam, you can actually talk to the external application.

In plain natural English, you can say, Go through all my PDFs and then pull out the top, you know, five best rated Java books.
That query or then translate to the following audio directory to analyze and find all the PDF titles right remote that are that PDF.
Go on to like maybe amazon.com and find a star rating for each one of them and sort all the stars and then show you the five top of books.
And I'll imagine writing code for that. Right? Is very painful. So no, there's no need for code.
You can basically just type whatever you want with it. Okay.
So therefore you can get high quality results back from some external knowledge base, one of which is knowledge graphs.
In fact, there are only two of them. I might as well tell you when you have an alarm, meaning this one lectured you pretty good,
you know you're talking to, but then you are telling it in your prompt.
Do not answer what I'm going to ask you with your own knowledge that that's what you start with.
Look, don't answer with what you know. It's in the prompt, okay? But instead should say use this external knowledge base.
That's what they call a rag see retrieval argument. That means take my prompt and run the prompt here, so to speak.
Then what a way to get results back. Chunks of results that might be used to answer my question.
Take all of those, add them to my prompt appointment based upon my prompts from a prompt becomes bigger.
But now the problem consists of really good answers that I would possibly want.
And then summarize all this for me. Using natural language, I'm waiting for your answer.
Imagine doing that right so that we are not using the actual words in the transformer.
So what was in the transformer is what people like open air and on chat many people
put in and that is not accurate in any specific domain to a general purpose.
As soon as you ask about U.S. tax law or anything, the [INAUDIBLE] right does not exoplanet's biology cannot possibly know.
But you can have unlimited external databases.
Only two kinds. One would be good. All things like relational databases.
You can have tables, you can have sequel format, relational tables,
or you can have things like MongoDB, JSON documents or free form documents like a whole blog,
all of medium.com articles, you know, for instance, can be over here or it can be a good old knowledge graph.
Well, so that's why this topic, I think is a great goal. Can a knowledge graph also your data?
The data is literally a graph. Okay. If you know about a program called neo 4G.
So once again, the database world, this has been popular for at least 1015 years just ends with Java and Neo stands for The Matrix movie.
So this whole bunch of Matrix aficionados that made this program, it's a graph database you store,
so you store knowledge stored data in a graph database that means you can store
the world Los Angeles one node like one vertex and ask a wire that says is a city.
And they have another circle here that says California.
Here, San Diego is a city in California, Los Angeles, San Diego, right distance between 200 miles.
So before you know it, you can take facts about the world and encourage them in this massive graph that can be endlessly big.
It can have trillions of northern Chile. Okay. Okay. So then that is what contains factual knowledge that we put in.
And now the system can be made to search that. And then tell me about Los Angeles.
Well, in Australia, Canada, you know, most populated cities in California,
if you visit L.A., you probably should go to San Diego to is only 20 miles away. So it seems to know like what?
Answer, right. But because it's searching the graphic but searching for knowledge graph.
So like, wonderful. All right. So USC has the school for it as well.
One of the biggest knowledge graphs is Wikipedia. Just to quickly tell you.
So Wikipedia, every article is a node like a vertex. Okay?
And every time you have hyperlinks, each article more, more black words.
So more than the black words that are blue links. Okay. Meaning every article links to so many articles.
So think of each article as a dot and all the blue links are wires to the whole.

Wikipedia is a blob. It's actually called wiki data and wiki data.
You can download it yourself because you want viewers for it where you can jump from Doctor, doctor, doctor.
You can magically fly through all the pages. That's not the same as clicking Wikipedia.
Okay, so there's so much I can tell you, but I'll tell you like as I think about them anyway, So USC has a very similar effort.
I used to live less than a mile from here. More west of there is Sevilla Marina, right by the beach.
So I used to live there. Okay, this one is on the Mindanao way.
Lincoln, if you guys have not been decided, please go to. ISI is 50 years old.
They are the Silicon Beach first tenant. Okay, before there was one called Silicon Beach.
Anyway, it's great. See that link maps? Okay. And then it's all about linking.
See this knowledge graph for business. So again, to, you know, come to 2024, what is happening is Google, when you do search,
it doesn't just simply use the inverted indexing that they showed. Right? They also use knowledge graph.
They started doing that about ten years ago. And Microsoft Bing Search also uses, sadly,
their own knowledge graph some data knowledge graphs allow them to okay because
the knowledge graph formats and the way this taught them are different.
But the fact is Los Angeles City in California, USC is one of the three big schools in Los Angeles.
They never changes. Okay? It doesn't matter if Microsoft is it, but for now, they're all different.
Someday they'll all be linked together. All right. So I'm going to start with that.
So much to say. So taxonomy, ontology, I define like all these terms.
Okay. And then word it just very briefly about word and all that word interfaces are human created graph of words.
That is all human created graph of words. In other words, there might be a word called vehicle.
There might be a subclass called Car.
And the wired up car is a type of vehicle, so much like a class hierarchy without the class definitions, just a class names.
Okay, Imagine making facts about the world in a massive graph like that.
Princeton made that it is actually called word in it. So the word and that then afterwards gave rise to something called Imagine It.
So people said, wow, we should for every word, go find 10,000 photographs or something.
Right. And label all of them with those words and then have a competition every year to
see who's in your network can label most of those words correctly for the images.
Okay. That is how basically modern Convolutional Neural network was born.
So CNN breakthrough came when suddenly the accuracy went from ten 20% recognition to like 80%.
The new world of deep learning was born from that. So word night is a precursor for imagine.
And imagine that. Like I said, it's a new revolution. Well, and then Wikipedia and Google has knowledge graph.
So does Facebook. By the way. Source Microsoft. There are many different knowledge graphs because it's a good idea.
Okay, so knowledge graph shall start with the notion of organizing the world.
That is our taxonomy. Okay, so you want to somehow classify things in the world.
As an example, if I asked you how do you classify things in the world? Well, that's such a general question.
Oh, come on. I assume anyway. You know, you can say everything in the world is either intangible or tangible, you know?
Right. Everything in the world. Intangible.
Intangible. Right. Tangible. Likewise.
Tangible. And I think what the [INAUDIBLE] they mean by that?
Well, you know, a wallet is tangible because I can touch it in my hand.
Somebody's height. Is the property that everybody has a height is intangible.
I cannot touch a property called height, rent, height or something that we made up.
I think I like to say this in class. I probably said it here to all of the data in the universe, any data in any database does not exist in the world.
It is there just because you measure it. Okay?
Otherwise, there's no such thing as the average height of students in this class of algebra or something.
There's nothing we just make a of basically. Okay. So then the tangible objects can be solids, liquids, gases.
And somebody would say, What about plasma? Plasma within solids?
I can have things like ductile solids and brittle solids. Like, what the [INAUDIBLE] do you mean?

Well, a brittle solid. If this was brittle for drop, it is going to shatter to ductile.
I drop it on shatter so I can go on and on. Right. That way of organizing the world.
It's called taxonomy. So we build a taxonomy of words so that we organize knowledge.
And I had the periodic table to big taxonomy of chemical elements.
Like there. Mendeleev had this cool idea. He said, Many elements are common properties.
I think I'll put them below each other. I study them as a group and hey, look, there's a box missing.
I think I'll go find that element. It's pretty neat, right? So that is all taxonomy.
You can read all this afterwards. Okay. It's about arrangement, classifications, about classification, ultimately.
Okay, so ontology also looks like a taxonomy, right?
But ontology is also about properties. Scientology means it's a taxonomy, meaning it looks like a graph.
Taxonomy looks like a graph. Rank taxonomy is a tree, by the way.
You hardly go back and you make it look. You know there's no loop.
What do you call a graph that has no look, It's a tree, but it's another name for a tree.
Does anybody know? Okay, suppose I draw a graph like this.
Okay. You would call it a tree, but I'm going to call it a graph.
Call it a graph. With this one really, really interesting property here.
What is it? In fact, why is it a tree and not a graph?
No cycle is exactly correct. So it's directed right?
Directed direction means the direction parent. So directed acyclic acyclic nor cycles graph.
So we call it a dag. Okay. These hierarchies are called dag sometimes, but they're based on a loosely called a tree quote.
So that's what ontologies are. You're basically go from top to bottom and then go back.
Okay. All right. So then this all looks like that. Okay. To satisfy the tree property.
But also, these are just now, right. It's not just simply this a vehicle.
This a car is a type of it's more like action verbs.
Event is directed by Los Angeles is located in USC, has population 40,000.
USC offers major computer science. So those wires become more like a relationship, you know, sad tables.
We call them relationship program between two different tables then that kind of have an abstract version,
meaning that's not actual people's names, right? Is called an ontology.
So ontology is a type of hierarchy which is all about relations, but it's more like an abstract class definition.
A quarter more s usually have more is usually will have a director so that you can have the director's name and say, you know, like Josh or something, you know, directed by Steven Spielberg.
No Jurassic Park. So likewise, actors, a movie has a particular actor actors or but there's no actual director's name mentioned.
But we can take this template and put actual data into it and make that become real.
Then you will have IMDB, right? So now IMDB, a real group of names.
And at that point I MTV would be called a knowledge graph.
So knowledge graphs come from these kinds of ontologies that is all about you first need to know what do you need to say between things?
And then they start filling it. If I say city name, City name is a city in state, for example, supporters say is a city and you know,
city and state, then I can make this be true by tens of thousands of things in the US.
And I can say, I don't know, New York City know city and state called New York.
Cedar Rapids is a city in Iowa, right? Glendale is a city in California.
So I can make that be true by actual real world examples. Right. Like this.
A class definition. I can make real objects, objects or an object objects by instantiating it.
So likewise, I can instantiate ontologies and get actual knowledge transfer to them.
Okay. That is really all. Like for example, this one, you know, it's a knowledge graph because it has very specific thing.
It's more abstract, right?
This is real in the sense that's an actual pattern number and there's a pattern about cloud computing and then likewise platform as a service,
you can go on the cloud and make an entire platform for doing like financial calculation.
Then that runs on the cloud. So it's related to like where somebody comments on cloud computing, right?

And this one is a paper actually MapReduce.
So MapReduce has Sanjay Gamow and Jeffrey Dean, the two authors that came up with MapReduce.
So that's an actual example of a MapReduce paper in the abstract, this might say paper title and then author,
first author name moderator But now it's not abstract, it's real, right?
Therefore, that's called a knowledge graph. All right. So the whole lecture is about creating these and using them.
Okay?
And these days, machine learning, you can use elements to automatically take a big book and convert what's in the book to piece of knowledge graph.
And obviously what's in the book might be a knowledge graph, right?
But then it might be that some of the knowledge in the knowledge graph, a knowledge might already be in a knowledge graph.
And when you say a merge, it simply leaves out what's already there.
In other words, say you already had this in a knowledge graph. Okay, you have a knowledge graph that already has this.
So then you have a new PDF file. You don't want it. It is too big.
You tell an Al-Alam, why don't you read it and see if you can add this knowledge graph.
The alarm says, Well, I found new notes, I'm going to add them.
So incredibly the knowledge graphs can be automatically calculated. Okay, that is pretty wild.
In fact, look here. Knowledge graphs.
Okay. L m knowledge graph generation.
Yeah. Hardy integrated knowledge graph.
Okay, this one is actually a lens to create knowledge graphs.
Not integrating them. Integrating them would be more about Iraq.
You already have a knowledge graph. You already have an alarm. You want to put them together.
But that's not what I'm saying. What I'm saying is you can actually use C like that.
Alarms for Automated knowledge Graph creation that. Hello.
Good morning. Good evening. Good afternoon. So this way to you don't have to know that it is going to be fully automated so I'm going to call this is
there's some kind of a text right movie that's usually called right under CSP and maybe this can be a PDF file,
can be a blog text, write all of that.
They can ingest like all of these and turn them into a knowledge graph and actually store them in near 4G and then right can be used somewhere else.
A chat agent can go and query like all of this now can answer the user.
Okay, it's extremely neat. Okay. Then here's how the dislike neophyte is writing to Aura because he worked okay, so it can go actually afterwards.
Anyway, it's a very neat idea that you can use something like a an alarm to create them.
Yeah. Okay. Yeah. This one is sort of tool augmentation.
So this is the one where the alarm queries the knowledge graph does not create it.
Okay? It's already created by somebody else. Anyway, there's all these combinations.
All right, I'm going to go on. So now we know in our knowledge graph, which is a very big deal.
Okay, so knowledge graph is used to do complex, unstructured,
D and all this right is usually unstructured, meaning the loose associations that we have in our heads.
The notion of a city is also abstract. Okay, we just call like a place city.
The notion of a state. Notion of a country is also abstract.
The notion of a political border is also abstract, as you can see in Gaza and Ukraine over and over is some imaginary line.
Okay, but people basically die over it. So then that is an example of unstructured, mostly unstructured to represent.
That is how you create the graph. After you create the graph, you follow the graph from note.
In order you search for things somebody might tell you Tell me the top ten most populated cities in California typed in plaintext.
Then it goes in the thing that talked about states and cities, right?
And finds a node called California and locates all of the wires that say some other city and is a city in a city and compares all the populations,
sorts them and takes the top ten and list them for you that way to answer your question.

Okay. So that way that might not be a Google page already indexed with that information, but it can go in the knowledge graph and run time.
Find it for you is more useful than document indexing. Again, because your answer is not in any documents.
This is exactly like a database. In a database in a table.
I have all this data, but then in the data there's no average up here, but I can run one single query image parentheses.
Japan. Suddenly I have a new piece of information that came from that existing piece of information.
Likewise, we can use knowledge graphs to search and then make the system come up with new answers that is not hardcoded at compile time in any page.
So dynamic search so useful. All right, so then that is all you have to represent of reason in expert systems.
It's the exact same idea. In expert systems, you would call this rule base a rule based like a document based knowledge base rule based.
So rule based is a bunch of rules. Rules simply mean this.
Each is called a rule. Let's call a rule, make a little base. And then the reasoning was called inference engine.
So rule base like this, rule base, inference engine, you can look at the similarities.
Okay, cool. Base inference engine. As soon as we see that we're back in the 1980s.
Expert system looking. Yeah, over and over.
Exact same thing. See that database role base and then you would query it.
But that's already created. And then your credit goes to an inference engine.
You ask, how do you put your question? It'll go and then do some searching here, come back with the names.
Okay. So over and over, similarly, I can go look at it again.
Same thing. You need the knowledge base first and then inference engine. So this is a representation.
That is the reasoning that will be like credibility. Okay. All right, so then that's cool.
And you're right. So if you go on Google and type things like, you know, ask dot com, is Elvis alive,
then it might as well go to a knowledge base because Elvis is a named entity.
So I want to introduce a concept called nurture. And the answer and this is somewhere in the future slider.
So when we get to that, we can easily skip it now. So notice that thing that is used to create things like knowledge graph named entity recognition.
So you have to name some entities that are already popular.
Some human being would make a massive list that'll contain names like Bill Gates, Elon Musk, Sam Altman, not I check.
Okay. If I become as famous as my name would get on them. But it doesn't matter to a name.
Famous names, you know Taylor Swift. Likewise. You also have famous places, you know, Shinjuku, Akihabara, you know Tokyo, right?
So then when you have again, it's people, places and things. Okay.
There's usually what I named people, places, things I think might be Microsoft Corporation, you name it.
Okay. So our system can have a big list of all of those.
Then you give it some unstructured text or your own video and say, Go word by word and every word.
Listen carefully and see if it matches any of these names. And if it does tag it, that is all it is.
So it can then automatically pull out nodes. Okay. Then each named entity will become a node.
So if you search for Bill Gates, already there's a knowledge graph with word Bill Gates in it.
And then there's also was born in Redmond.
Now CEO of Microsoft Corp is worth in $100 Billion note right then that way your search for Bill Gates.
Tell me about Bill Gates. You will then summarize like all the information that surrounds the Bill Gates note.
Okay. And then it appears intelligent one. You know, I have a biography of what Bill Gates generated, but people have tried this.
Okay. So if you go and ask for your own biography of my biography, it makes empire [INAUDIBLE] up.
It's complete B.S. So you should not let Dalai Lama answer that question directly yesterday.
I should say go to a knowledge graph and then find out what's in them through the knowledge graph.
Does not have Macquarie come back and tell me. I don't know.
You type all that in the prompt, take it out the little clock. So that way the knowledge graph does not know what you want.
You'll say, Sorry, I don't know how to answer that. You should be able to tell somebody.
I don't know the answer to that. You shouldn't police somebody by randomly giving them crap just because you can.
Okay. All right. So then depending on how what knowledge graph was useful, all of these hopefully it'll say the right thing

again.
Is Elvis alive? You know, he basically passed away except for conspiracy theories.
So Ask.com, it says mostly as Bing said.
Yes, maybe that's all idiotic.
Okay. He died of like, sadly, drug interaction, overdose. Okay, So then Google says, no, he's not alive.
So it's so strange that, you know, Wolfram, by the way, is a Mathematica search engine.
Wolfram is gosh darn smart. Usually you ask math calculations about PI, for instance.
Okay. But you can ask Google. So you can ask. Wolfram is always alive and thankfully it's more like Google.
So in other words, based on how the quality of your knowledge graph is, oh my God, lawyer shows up.
I just like. Just kidding. Just kidding. So that's why we're ready.
Okay, so then you have to use a sentence, which is pretty sad,
but I lied a little bit to you and I told you it used knowledge graph, so it actually did not use knowledge graphs.
If you don't use knowledge graph, if you only go by documents, okay, document indexing and sadly, you get this crazy, you know, type.
And in the real world, this is not either he's alive or he's not alive.
There's no Schrödinger's cat. There's no single simultaneous superposition.
Okay? And the truth is, he actually died, by the way. So it's like, based here.
I'm okay. I'll tell you then. I know what you think about it. Probably. All right, cool.
So you must use knowledge. Graph is the point. You must use knowledge graphs.
So knowledge graphs came from a I, I want to put in a personal plug, not a plug.
More like a little, you know, data point. I work in the world's largest knowledge graph project.
The world's largest. It was the largest. It will always be the largest.
It can ever be even bigger. There's no charge. It can do this.
Okay. It is called the psych Project. Crazy.
Oh, we said it only one year, but the one year was enough for me. I didn't need to be.
But the project actually ran ten years. Okay? The project ran from 1984 to 1994.
The project's goal was to create the world's largest knowledge graph, but not any old knowledge.
Graph and knowledge graph did not have simple facts like Bill Gates and I was born on that,
and knowledge graph was more about common sense in the world. Common sense would tell you that a solid object will not go through a solid object.
We could represent that in the graphic, but could also represent the fact that some objects, when you drop them, they'll not break, not break.
Other objects when you break them might break. Look, we call it brittle.
So brittle, not brittle. So we know all this from common sense that we live in the world like liquid if you're trapped upside down but poor.
She cannot walk around with an open container upside down. Even the smallest kid in the world would know that, right?
Because the little milk bottle that the kid had, all the milk is done in the trash.
You learn just by experiencing the world.
But we have to represent all of those bits of knowledge and knowledge craft and our scholastic knowledge base.
Okay. Yeah. And we constructed a programing language on top of Prolog.
Call cycle cycle influence.
Psych project. This had 30 million rolls in it.
30 million, right? 30 million nodes and wires between them.
Just completely crazy like all of this. Right? I'll just say our psych psych project ontology.
I'll just show you, like, an example of ontology. And then it applied the ontology to real world.
Oh, my God. Look at this. I was part of all of this. Meaning I wrote code or I wrote some code to search through all of these.
What I'm going to show you. That's what actually ontology looks like.
Intangible, intangible, individual and intangible individual has to be a ghost.
Okay. An angel. Ghost. A demon of God. Satan.
Satan probably is an intangible, you know, entity is crazy.
Okay, so that single game spatial thing, some things are spatial things are that things are not.
Time can be not just one thing call time. Time is usually interval like this class in the last 3 hours.

Right? So crazy, right? We had like so many of these and then ran searches on top of them.
We try to see we try to see if the system would actually exhibit common sense.
The whole purpose of the psych project was to see if we can replace human common sense with this
big 20 million strong rule base to see if it can have general purpose knowledge like you and me.
If I ask it, you know, if I bend, it would have been to say a settlement.
Why? Well, it's made of flexible plastic. So what's the goal?
Okay, I can tell you the goal after ten years, radically, utterly, horribly, disgustingly failed.
So the whole psych project was like a stunning failure, not a technical failure.
Meaning it all work again, but the idea that you can distill common sense from our lives and put them in like a knowledge base.
Okay. By the way, welcome to 2024. The idea is still B.S. So no, the new version of that is called Al-Alam.
The new version of that is called Multimodal Al-Alam. The even more new version of that is take a body robot and put them in its head and then see it.
Talk to you. That's a video of somebody going and asking your questions, right?
There's a bunch of dishwasher plates in Iraq. And the guy says, what all this place called the robot says it goes in the dishwasher.
That puts in it all sound very cool and everything. Okay. But I can tell you is B.S. It's like plus plus B.S.
Okay. It will not work. Time will tell. Yeah, I'm pretty pessimistic, negative about all of those because I've seen it.
I know what it is. All right, so that aside, that's a lot of money to be made.
Go. So then knowledge graph does the things I told you.
Yes, a knowledge base and All right. There's all kinds of knowledge basis.
You know, even Apple has one, IBM has one and then so does Facebook.
So does Microsoft. The all your knowledge base is the very easy to make.
Okay. And we can all still Wikipedia by the way Wikipedia.
So like a free open source knowledge, but you can only take it and build on top of that call.
So then an object model is just simply, you know, the actual representation.
So now it gets into things like classes.
You know, for example, a vehicle is a transportation device is a class that can be land vehicle, water vehicle, air vehicle.
And a land vehicle like in cars SUV is, you know, bicycle skateboards.
So I can build this taxonomy. And so then we call it things like an object model.
So those are all objects like vehicles, cars. Great. And then you can have things like again, parent child, these are things, you know,
from programing, okay, a superclass is on the top subclass of class child.
Child can operate, can go down.
So as you go down becomes more and more narrow like in a car sports car in our sedan two door for our skip going hatchback.
But the more you go up it's more broad like transportation device you know cool so knowledge basis
then some Yeah like this you know any kind of relationship at all about humans about locations,
about things you know most cars have gas engines you know, and electric motors, for example.
So those are all facts about the world, right? You make them all into one big graph and you call it a knowledge base, like a graph like them.
Then search engines will search. And it word that you type in telling me about electric vehicles,
it might as well go to a knowledge graph and find all the notes about EVs and summarize all the information in there.
And also today as we speak, is using actual document index as well.
But the point is, over time, next two years, three years, gradually there are URLs that they give you,
you know selector So the US CV market is supposed to be pretty bad, right?
And on all that you see we make it. So that word came from searching like all these links came from actual like this, right.
This, the homework that you did.
But in the future it might not have the word TV, It might have a the word electric vehicle in the knowledge graph somewhere else.
There might be an equivalence. It says anytime somebody says electric vehicle, you can say even race versa.
So if you type EV, you will find pages that say the word electric vehicle, you know, like amortization.

Okay, Somebody made an equivalence. So it can use words that are not in the document than say a question.
But right now that's purely keyword search. But slowly the keyword searches are being augmented by knowledge graph search.
Okay, cool. Okay. Yeah. I can see here, when you say Picasso, you know, I'm just like, Picasso is going to tell you a lot about Picasso.
All my three topics today are somewhat linked together because you see that about because I look right here.
This all came possibly from a knowledge graph. There was no document, actually.
That's why there's no link there probably came from knowledge graph.
But the knowledge graph has those nodes was born in and then it might say Malaga, Spain was born during that day.
But it and likewise that you know so I can take the knowledge graph and give it in the form of words.
I can next up summarize it as like a paragraph. Your next step really do an intermediate.
The Speaker to gradually we can do more and more. But the idea is that they using it also this kind of highlighting something.
I just search for one word he started giving me actually because of paintings.
Right. That is called a snippet. It's called a rich snippet.
Snippet snippet means a piece of search that is highly useful to you.
When Google first started 20 years ago, 30 years ago, there was no snippets.
The word Picasso gave you a bunch of URLs that are all somewhere hardwired.
Picasso like your work. But gradually people got bored and said, Why do you give me at all links?
Can you just give me some facts? That is how the knowledge graph started to come together.
And they invented this new thing called a snippet. It's actually called a rich snippet.
Rich means not text like multimodal. So anything that is rich is images and also videos for the name of a movie type consultant are for right?
Don't just tell me an article about how DreamWorks might come to mind or for you to show me movie trailers.
Actually, there are movie times in movie theaters, right?
Actually do it. Or now that is called a rich snippet, so snippets are highly useful.
I'll just take care of before and see what happens. The state knew that carefully.
As for the short time to write, all this automatically pulled from Fandango and you know, AMC like all kinds of sites that have all this information.
Cool right? And it seems nowhere we are throw a lifeline is only showing you things that are very close.
Okay. Okay. So that is called a snippet will come back to snippets over here and slowly walk into like random little situps here and there.
We're getting there. All right. So a knowledge graph for Elvis. Again, we want to like, worry about every single one of them.
All these are facts known to people. Okay. So humans can type this.
And Elvis's wife, she's still alive. Priscilla is married to and backwards.
If you go to her node, it might say, you know what's married to Elvis Presley?
Why? It was you know, actually there should be was the reason there's not.
Is that right? So you can have again what what do you have here things like links to we're married to has given name you know Elvis
is actual name she was born on date that is where the birthday comes from has family name as Presley has gender, male, male, female, non-binary and then has child marry.
Lisa marie is a child, so she's in there.
So when you zoom in on her, she'll have a note that says, Has father Elvis Presley has mother, you know, Priscilla and so on.
Right. It's very cool. So you can see how a knowledge graph can be created, right?
All right. So then that's what you do. You said, tell me something.
What? Elvis will tell you something like this. The first and greatest American rock and roll star died yesterday at the age of.
That's weird. Hopefully tragic. Really shouldn't tell you that, right? Got a kid would believe that.
Okay. All right. So types of knowledge base.
Again, you know, you can have like all kinds of knowledge.
That knowledge is not just one thing in the world, you know, in fact, search engines and in fact, in my mind, I spoke

earlier.
It's only good for a small subset of those things.
You can basically Google looking like, you know, what type of programing languages, C++, you might use,
an object oriented language, what type of programing language is Haskell Functional programing language.
That's a very simple object, the fact that a lot of people should know.
So today's still as good at coming up with just those you ask it more critical things like why are you into programing?
What are humans get from programing? It cannot answer all of the okay, just cannot.
So I want bashed too much but honestly so attacks taxonomic knowledge or taxonomic knowledge.
So taxonomy is about paper classification. So how do you classify Elvis Presley?
What is an American sample? How do you classify? He had a pretty low voice, ladies and gentlemen.
So there's like a baritone voice, right size type baritone. And then so how do you classify an American singer?
When I got a subclass of singer, what kind of a person is a singer? A singer's an entertainer.
What kind of person is the entertainer? What a human being. Okay. What do you think?
Hopefully robots can entertain us someday and entertain more than so that is the idea of taxonomy factual knowledge.
So who sang All Shook Up? So that's Elvis Presley singing.
And then likewise in a factual Where were you born? Tupelo, Mississippi.
Quote I've been married to, you know. Good idea, twin brother.
So I actually didn't know that. So interesting, right?
After many years and after Elvis passed away, Jessie might come out of the woodworks and say, hey, you know, I'm actually a twin.
I didn't tell you. Whoa, what the [INAUDIBLE], you know? Tell me more about Elvis. And you have a new note called his twin brother.
Okay, then for him, it also has twin brother, Elvis, because it goes both ways, obviously.
All right. That is interesting. He partly was Native American.
Again, a lot of people would know that's a Cherokee Indian. So what is Cherokee, a type of Native American?
You can just go on and on. Right. So it's a very cool idea. All right.
So hierarchy, again, is a directed labeled cyclical.
You know, ontology should not be cyclical.
It actually should say acyclic graph. I truly don't know why it's a cyclical.
Cycles are bad. Okay, yeah, please make it a cyclical. He said Dog, we call it a dog directory segment.
Yeah, you don't even have to call it labels because dogs will have labels anyway.
In hope, you know, just from programing. So that parents passed their properties down to a child.
So a vehicle has a property called weight. How heavy is then?
If a car set up a vehicle, I expect the property. Every car should have a slot car.
Wait, I want an RV. The car is right. So parents pass properties to their children.
Children can also have additional properties that parents don't have and they can pass it onto their children.
It's literally call in public inheritance. I have a class hierarchy.
It is public inheritance. All the parents, members and matters will go to a child and that that inheritance also public will go to the grandchildren.
And the only way to make it private, it'll start propagating right here.
It's all assumed to be public. Okay. Inheritance, that is all.
So assumes of properties, meaning that that's a class definition and each object will have then the members like a car.
I might have a car cloudy with a certain vehicle identification number and the Audi
VIN number would have a weight property because the car class had a weight property,
then that's an individual. Okay, okay. And then that's all saying it's basically about is a it's called is a type of relationship.
Right. And programing a child is a parent, a car is a vehicle, a vehicle is a transportation does not has a hazard means components, you know,
like a car engine might have in all kinds of components in it does not a Bahasa it's all
about is all right that is all over and over symptom general at the top spirit at the bottom.
You know as one example right Speedo is a specific type of monkey you can probably in a zoo and sofrito is a type of monkey and monkeys.
An animal. Animal is a primate. Primate might be a moving living species as a like a funnel, what you call a funnel as opposed to flora,

which is like trees and, you know, mushrooms and grass that don't move. Okay.
And so then everybody has values. Suppose a primate has a value called expected lifetime or something, right?
So every, you know, kind of animal would have some expected left and then animal will have an expected lifetime soil monkey.
And so then you would not fit over probably 50 years. Okay. And where did that come from?
Like how can you and ask how long we have to live? Because every primate has the notion of life.
You know, in fact, if you if you make this to be a flow of fauna and make this to be a flora,
that can be unified into thing called life, because then all the other objects are not alive.
So I can go here and say everything in the universe is either live or not alive.
Then somebody would say, What about a virus? Well, you know, pick one. Okay, something in between.
So I'll probably call it not alive. Viruses actually not alive, okay.
But they know how to propagate tonight. All right, So then the top values and go to the bottom level.
So these values, plus these values would go here and those values would be applied to you see the differential, right?
This is an abstract class hierarchy. This an actual instance that means the instance like an actual object as one.
Yes, like monkey feet or equal to new monkey parentheses is actually what's happening.
So therefore, this object will then have all these properties because it's public inheritance.
Public inheritance quo in all this. Okay, okay.
Can completely not read all this because that's all it is. So and then the knowledge graph becomes an extremely powerful object,
meaning that's a thing we can create in the world and it will fully solve hallucination problems.
Ill alarm, hallucination problems. I want to tell you. And then we can come back here.
El alarm, hallucination. Make a poor hallucination.
Solved. Solved by how?
Select three point direct strategy to stop hallucination.
Bad answers. Okay, you can send away for that.
Okay. Because you go on searching the knowledge graph.
Senator about hallucination. Mm hmm.
Maybe this one. Okay. So look how cold it is.
This right here is the reason why knowledge graphs again, are nice.
Because you see here, supposing you have a document where real knowledge lies and you come up with a pretty cool neural
network with lots of training on those kinds of documents so that it knows what's in the document.
And to give it a brand new document, like a brand new image, you want to get a label very similar.
Then it knows how to properly embed the document into vectors. Okay, exactly.
The multidimensional vector we talked a lot about. So these all become points in multidimensional space.
Okay, so then leave that aside. Just leave it right there. You ask the Dalai Lama question.
You asked that question, but I'll tell them the elements right here.
Don't answer my question directly, but instead take my question search here in the vector, do a search,
come back with some answers, add it to my query, add it to my prompt, and then the answer that you need is already here.
Now just simply answer me. It's so easy.
So stop del them from answering. This is the external augmentation because this is not external is internal.
The external part is here.
So this can be a vector database or it can be a standard database like this, like relational table, or it can be a knowledge graph.
Okay. Yeah. By the way, the knowledge graphs, these are all not different things.
I can take tables and I can take knowledge graphs and recognize them and then they all look like two databases.
You know what? Nvidia is doing this pretty amazing.
Nvidia has a brand new chip design group where they want to build chips that have never been in the world before yet so far,
which is hardware acceleration like they've done for a standard machine learning vector databases.
Wow. So vector search chip, which means it'll then speed up all this embedding.
And this can be very slow, by the way, and also speed up the query. Wow.

So brand new group. Okay. Good. All right, so then back here, how do you how do you store them?
Okay. So you need to actually be able to represent. Right? So suppose I tell you just this relationship.
Elvis was born in Tupelo. Okay, that's pretty simple.
You can visually draw a little note, like to write down a few examples like that, or you can do this pretty cool thing called a triple store.
Triple store. So a triple store is this I'm going to go here.
Subject. Object. Subject object kind of by this thing.
Call predicate. So one way to store this is subject Come up pretty good.
Come object. Okay. Or you can even do predicate parentheses, subject object and store them in so many different ways.
That is what is on the right. And so in the then what it is is USC is a school in Los Angeles.
That's all. So that is my subject. That is my predicate that's mounting.
It's always two at a time, kind of a bilateral relationship where the entire universe,
literally of knowledge can be three pieces at a time, assembled and gradually fuzed in there and more things.
What do you mean by that?
Suppose I show a thing here called California, and suddenly this object called L.A., right will now become a subject because California.
Sorry. Los Angeles is a city and is a city in California.
And likewise, California is now an object. Right.
But it's going to become a subject because you can say is a state and is a state and us and so on.
So that way I can represent the surplus. Right? That is what on the right hand side, amazingly, we can use a relational table like musical cycle.
Like what? Order Oracle to actually represent them. Make three columns in the subject column.
Pretty good column. Object. Column. And literally take this relationship.
Elvis Subject Predicate. Born in Tupelo.
Object. Wow. You can fill a three column table, but trillion rows and rows together will become the knowledge base.
And so the search engine can do a classic sequel query.
You can say, Tell me a subject where predicate equal to type and object equals a singer.
Then it'll go and find every person who's labeled a singer. You can write second queries, okay.
You can write many kinds of queries, some what is called native queries.
There are standard pretty good coding languages or you can always query using sequel.
So I'm telling you more things than you need. What do I not write like this one said type Elvis.
That's what I wrote there. You can do predicate parentheses, subject commands.
Please don't try and swap them will be highly confusing. Please keep subject first.
An object later can help preserve the relationship. Left. Right. All right.
So then that is all. And you have standard formats in the world.
There's a format that's actually there to a standard called the resource description format.
RDF sometimes declared to already have triple and RDF triple.
So next HTML file format. I make a little bit extra Mel extra Mel file to say that Elvis was born in Tupelo.
Okay. I can actually quickly show that I don't have that.
Oh, there it is, actually. So this one is abstract, right?
So Bob is a person, but this can be an actual similar tag, and then that'll become an actual XML file.
One little snippet and then that shows Bob as a person. So to use this example, suppose Bob is a person.
Obviously Bob was a person. Bob was a friend of Alice.
So Alice and Bob are friends again. Likewise. Alice is also a friend of Bob.
It goes both ways. Bob was born on July 1991, and then Bob was interested in the Mona Lisa.
So that means you can say you can ask the system what else want to go with Bob, you know, to see Mona Lisa.
And then you can say, Yeah, but it's just too much to ask. Well, why did you say yes?
Because Alice and Baba friends. So hopefully Bob wants to do something that I would follow along, you know.
And Bob happens to like Mona Lisa. So we think Alice will go with him and so on.
Right. All right. So then that is we can collect whole node stuff.
You know, they're all like ultimately nodes, right? Like, we're headed in the previous slide.
And even query them are going to query them. We can do things like signal or use a language call sparkle like, you

know, my hair sparkles glittering.
So it's called sparkle spark, Sparkle language Sparkle.
But Sparkle is not sequel. So triple star acquired language.
Okay, maybe underneath it turns into sequel.
There's a compiler for it, but you can write at a high level description a question like answer me something like,
Is it true that Bob is interested in Mona Lisa and it'll find a normal Bob Marley sound and find a pretty good girl is interested in?
If it does not find his predicate, it'll come back and say, Not if you ask.
Is Alice interested in Mona Lisa? There's no wire from Alice, Mona Lisa this time.
And come back and say, No, she's not. But if you ask, could she be interested?
It might say yes. Why? Well, because Barbara is interested. Alice and Baba friends, you know, a little bit more inference.
Okay. Quote. And then, yeah, this notion of a game and then you know, this all just about you asking something is always alive.
Then you can actually go and turn that into a pretty good and then go into some knowledge graph and then come back with all these documents tables.
You know, a table can store triples, probably can find some out answer in a more attractive right kind of way and become an answer the query.
That's all it is. All right. So now we get more and more into how you're represented.
I might skip somebody a little bit too much. The idea is.
And then the song. Okay, so the singer. Singer is a person person as a resource.
It's kind of weird, actually, huh? Okay, then. Now again, people place things.
It's always. People place things. People I think might be Elvis.
And I love to play, you know, pretty fancy guitars. Okay. Okay so multi graph.
Right. So now multi graph might be like a node has.
Okay this node right. Cannot manipulation say Elvis was born in Tupelo.
Elvis attended church so Passover. So I called I attended church in Tupelo so two nodes don't have to only have one relationship
obviously can have many many many relationship where the two of you might be married to each other.
Two of you might be dating. Two who might be neighbors in an apartment.
Two of you might have gone to the same school in undergrad. So between any two of you, Right.
I can have all kinds of wires that go between two of you. So it's called a multi graph because there are multi arcs between any two nodes.
Okay. Okay. It's all easy stuff. And then you have this notion of synonyms.
Again, humans have to do this. So you would know from the past that Elvis was called the king.
Then if you say, is the king alive, then it is still come back and tell you it not alive because it knows that Elvis.
Okay. Someone has to externally maintain all the relationship. That is all.
I'm going to skip a lot of this, right? Is pretty simple. All right. Yeah.
See, now, again, the word Elvis, right? If you then say somebody else was called Elvis.
And unfortunately, then if you say was Elvis alive, it might come back and say the person is dead.
But one day a friend is called Elvis, you know. But you're asking about your friend. So there can be ambiguities in all of this.
By the way, Chad suffers from all these ambiguities also. Okay.
Because they're simply playing with words. All right. So how do you build a knowledge graph?
You need to have classes, which is the ontology.
Then you have to have actual instances, which is people like, you know, Madonna, you know, Taylor Swift, cool Beyoncé and Elvis Presley.
And Eagles. All right. So just more complimentary information.
All right. So this one is you can merge them.
See, any organization can write little and program to go to any document and start creating all these little nodes.
Right. So bubbles. So I create my sub graph over here.
You create a sub graph. It is a partition.
So they're not the same sub graph, but somebody can fuze them together knowing that the word Elvis at the top or even from photographs,

this picture looks like the same as that picture 100%.
So fuze them so that it says married and then it says, you know better like why does that look so much like that?
I'll delete one of them because I know they are the same.
When you do that, that's called separating the Confederate, the four knowledge graphs and even make a bigger knowledge graph out of all of them.
So in the federate, you have to have some kind of an equivalence defined by somebody.
So you have to know that they usually have a year, a month and a day.
And if you only have the year, it's the same year. So we don't need this stuff like that.
Okay. Look what is equivalent to what are the word cancer in the word tumor, the equivalent.
So that if somebody searches for cancer, I should give them tumor documents and vice versa.
So our doctor would know that. Right. So once a human knows what is equal into what, we can,
then fuze different knowledge graphs and make a bigger knowledge graph and hopefully for all of humanity have just one of them.
Okay. Okay. Otherwise, see, there's an Elvis PDA and a separate Wikipedia page.
What they're saying is they're two slightly different pieces of information, but anybody can combine them any time and make a bigger graph.
Okay? Yeah. So once again, who's the spouse or the guitar player?
So the who? Well, when you say guitar player, what do you mean? Well, that's the guitar player.
Wow. That person is same person. Married spouse should do a little jump between all these notes, Right?
Answer the question. So we call it inferencing. Okay. Logical inference.
We also call it symbolic reasoning because they can represent all of these into little logical relations.
Okay. Like P implies q pr, q pin, not q.
Some of you may know what I'm talking about. Who knows where the horn classes?
Yeah. So it's up to you. Yeah.
So you can represent these kinds of relations in a very interesting, you know, using like an or not basically in little expressions.
Okay, So we call them horn classes and then the horn classes can be actually used to reason over things.
So if you want, you can look this up. Okay. Horn cluster.
John was a computer scientist, technological person like this logical distinction of letters.
So I don't say these things, you know, because the class can actually get very dry.
And I spend half an hour on this. I mean, it's cool and everything, but, you know, it doesn't pay.
Our bills are far more useful to tell you. But I can point them at you.
But anyway, these are pretty neat, actually, given a bunch of in a horn classes like this,
an inference engine can reduce all of them to, you know, one of the guess or not for example of a boolean.
True or false. All right, so what else was race on?
Right? One and a half hours. Oh, my God. Yes. You know, recording.
Excuse things, right?
If you know that, you know some kind of similarity between, like, I don't know, pictures you can see, like, right there, you've two pictures.
You're an animal. I can tell you in the future. Look at the eyes. Okay. Look at the long bridge of the nose.
I think these are the same photograph. I look at the hair and a slick hair going backwards. Then.
Then I can decide it's the same person, same oneness, and one part of a knowledge graph.
Another is in a separate knowledge graph that can be used to fuze them together.
Okay. Yeah. So you have to make at some point judgment to say yes, I think they're the same or No, they're not same.
By the way, there's something called I want to tell you the end should ification that's in your word
and [INAUDIBLE] ification over the internet and then it should ification of the internet.
It's actually pretty real. So very soon. In a few years, basically.
Things like these to find out what is similar to what is going to become damn difficult because a generator would produce even more crap.
It would produce a mix between Paul McCartney and Elvis Presley and then flood the Internet with the word, you know, Elvis Presley.
And so then suddenly another guy trying to find if they're all the same would have a much harder time.
In scientific literature, already, there's references that are people people are publishing without checking of the

references, real or not.

Because in a producer reference, when an actual paper reviewer goes to see, Hey, what is this paper?

I've never seen this in my life before. The paper does not exist. And you think this is B.S., right?

This April Fools joke. This happening in today's scientific literature. Already so sad, right?

Oh, my God. So the Journal of the conference cannot blindly publish something again.

The reviewer now has a much more important job to say. That is B.S.

Why did you put that in? Oh, so what if the reviewer does not know at some point?

Oh, we're all screwed pretty soon. Okay, so in a sense, again, you applied like I told you all this, right?

Therefore, you put facts in a knowledge base and run some kind of reasoning to answer ultimately a question here in the search engine query.

Okay, So far, our training is you go from A plus, B, B implies C, and then implicit you go in the direction of the arrow from subject to object.

You can also go backwards. Okay. For an object, you can find what subjects match the condition that is called backward chaining.

You can do both actually you going forwards and backwards. Okay, so it's all about and more like, you know then.

So check it. Quote. Yes. You said these rules are the ones that are like form classes.

Okay. So find a bunch of horn classes that are basically coming from your searching subject, pretty good object.

And then combine all the horn classes into one final result that you want all.

Yeah. You know, sometimes it might be a new fact that you find, you know, like, there might be disparate pieces of information.

You put it all together and you find, Wow, I had no idea Lisa marie Presley was interested in butterflies.

Okay, that's a cool new thing that you might actually find new, meaning new just by joining.

And then you can add that to it. The knowledge base is one more piece of knowledge. This is where you wonder you still am being creative.

You know, it found something that humans didn't find, but I would push back and say, that's B.S. okay.

It's not that we didn't know Lisa marie Presley, you know, like butterflies or not, most of us did not know it was in one document.

But it's not something that they came up with. It's not creative. Okay? I cannot be creative, actually, in a way.

Okay. So moving on, Semantic Network.

Yeah. Again, that's also what we that grass is sometimes called a semantic network.

The word semantics means meaning by the same meaning network.

Why is it a meaning network? Because we humans knew exactly where to put where and they came from the real world.

You know, Bill Gates, for example, was not born in Los Angeles. Okay. You can go back to 1963 for that to happen or didn't happen.

So we put that in there. And so there's no B.S. There's no there's no hallucination.

All right. So over and over, same thing, right then. No words are actual like nouns in a way.

And these are the verbs, things like, you know, it's a city and has a population and you labeled that kind of everything that we can reason.

And this all makes sense to you, right? Okay, one more example.

Again, it's an actual knowledge graph. Wow. Water.

Okay, so snow is a form of water, and ice is also a form of water.

And the new makes snowman from snow. Typically you don't make snowmen from ice.

Okay, guys, it's too hard. You cannot them together.

If you talk to some Eskimo Indians, they would tell you that is somewhat B.S. It's oversimplification.

What do you mean? Because, you know, we have 20 words for ice.

He say what? There's 20 kinds of ice. If you come with me, I'll show you.

Because in the culture in the world, yes, there are 20 kinds of ice.

I'm not making that up. So you and I would just naively you're going to just call it ice.

Okay. Yeah. So in other words, all these are shades of meaning to have no in Oklahoma, but we still have to start somewhere.

So we just simply put that in. Okay. Okay. Eskimo.

We're not Eskimo. Types of snow.

Yep. See.

Look at that. Cannock. Cannock. Kind of look like they're all different.
50 Swiss know you know it even more not okay.
They live near the poles. No, they're. They're. All right. So therefore, you know, these are margin simplifications, but it still works.
All right. So is a link is this how a parent is able to get things down to because a child as a parent.
Okay. All right. This classic example, again, someone Hassan, you know, right.
So now Hasan Hasan lives in Jordan and watch this record a month and then I son as a person and that person is a mammal mammals,
an animal life form in order we call. Then here are just more of these.
So this one is no more general, Right? So this one is what you might call an ontology, because it's not an actual bird, you know,
although at this point it might become a knowledge base because it's a class and this are an actual object.
Instantiated. You understand the difference, right? Abstract.
Specific class definition object over and over.
Same thing. Ha!
We'll introduce word night and we'll take a break then. On again is simply a form of knowledge graph.
It's a hierarchical knowledge graph made by humans in Princeton to basically make a large
knowledge graph that they can search through and see how well the inference engines work.
Okay,
Meaning can we infer from things we already know because we put them by hand and ordinate and then the image version of that is called Imagine that.
What do we challenging networks? Can you identify any labels with more than 80% accuracy, yes or no?
So when it is a lexical, that means a word word database with many classes of classify classes getting standard class definition,
select this 82,000 class as well. Actually 82,000 different kinds of things in the world.
For example, rocks, books, helicopters, roads, musical instruments, fruits,
types of houses, types of building architecture, types of paintings on and on and on.
Right. I'll show you that. For example, Princeton.
Yeah, like this. So since that synonyms, you know,
the group like similar words together and also definitions and usage this very
cool right is a good way to learn language for people that speak the language.
And so the idea is can you train and learn the language good And then so one 18,000 class labels.
So when you have so many actual classes,
when the instance them you know then how many instances what you have one name in different and otherwise there might be ten types of butterflies.
Eight types are animals.
They're all real, you know, like tiger, cat, dog would call and then you make a database out of it and train like and as an example,
that is what actually it looks like when you go search where that. We're not here.
You can browse through. Yeah, there's a thing called use word in it all.
And. Exactly. I was going to say this Browns word and. Oh, no way.
Oh, look, the rotor in Pearl. Okay. What should we search for?
Corgi Corgis are very funny. Cute little dogs, right? Oh, so Corgi.
It found this corgi. Sue. US Corgi is a type of corgi.
And then. Yeah.
Sister Tom. Well, sir, Corgis, you know, you can call all these right there.
Look pretty old, by the way. So it's a lap dog. Qatari hunting dog, you know.
So many. Right. So you can click on any one of them will lead you to more.
So it's a big hierarchy, huh? Mexican hairless Chihuahua hunting dog.
Well, then you can search for it. Then it found a system called hunting dog, right?
No idea what's going to happen with this. Mm hmm. So I don't know what all this.
Right. I've not heard this in a while. Okay. You can go and look at it, but I just know that you can go and search for, like, all these hierarchies.

That's pretty much it. And then, if for any word, it'll actually show you that the definition for the word.

Okay, So then the idea would be that you would use this non hierarchy non knowledge base to see if it can become intelligent or not.

By the way, charge repeated does not do this. Okay. In charge of beauty you don't use word in it.

So what is the difference like? What do you think? How would you classify how Egypt is trained on imported open?

I don't think. Didn't use ordinary. Instead, what did they do?

Anybody. As briefly, two very different things we can do again to the third one called Reinforcement learning.

I'm going to leave that out. But there's two things we can do. Only two things, by the way.

Two things. Train air to make air a one.

We can start with our own knowledge graph, which is symbolic graph oriented subject predicted object is our ontology hierarchy knowledge base.

What everything I told you and train the air based on that rest of ground truth.

Or we can take raw actually we can take data.

First of all, take data. Data is not this.

So data simply means I list everything that I know about animals.

In fact, all in just entire encyclopedia is about animals. Okay? And either label it label it label.

But the labeling is still not a very complex knowledge graph structure.

Knowledge graph structure you might have a dog is a canine animal. In here you would just label a dog directly to be an animal, not to be a kind of.

So it's somewhat like a knowledge graph, but much more like simpler than a knowledge graph,

but even worse, unlabeled don't even label anything, just simply unlabeled.

So then this loosely, we can call this part L and loosely we can call all of this part symbolic.

AA Well, sometimes called the yeah, symbolic.

And this you can also call Nero, by the way, because it looks like neural networks, right?

So we call it neuro symbolic when you combine them. Yeah. So this one is unsupervised learning.

This one is supervised learning. In supervised learning, the supervision comes from your labeling in unsupervised.

You don't label anything.

So the label literally means take a whole blog about making like, pizza and then annotating every word about what the words mean.

Right? But then that's exactly what open air does not do.

They go to some random blog and get the whole pizza recipe, and that becomes part of the transformer.

Okay. Just embeds it. Then you think, where the [INAUDIBLE] does the knowledge come from?

Okay, here. The knowledge comes from us.

Obviously carefully constructing hierarchies about the whole world here.

The knowledge comes from you labeling this to be a water bottle, labeling to be a global labeling this to be a glove.

Okay. In billions images. And then if I show it like a new thing, it's a glasses case because somebody went and labeled all of them 10,000 times.

Okay. But the question is, where the [INAUDIBLE] does the knowledge here come from?

What do you think? Where does the knowledge lie come from?

In unsupervised learning. Unsupervised machine learning, which is basically what Jim and I tried to predict.

Cloud Mistral. All of those are based on sniffing out crackers.

Open source, by the way. I'm sorry.

I mean, kind of. Yes. Yeah, you're right. You're definitely on the right track.

But a more general answer. You pre clustering.

Before you run any algorithm. Where's the knowledge?

You're actually pretty, you know. Tell me about Los Angeles. It seems to tell you there's a popular beach called Santa monica Beach.

And so that there's Venice Beach, you know it, which seems now so hard as it now, so to speak.

Like, where's the knowledge? Anybody can answer. Yeah.

Still on the right track but an even simpler. And so you're right. Right.

I'm sorry. What similarities? You know. No, You got a little bit called kind of.

You're also somewhat on the right track. But all I'm asking is, where is the knowledge?

In here. The knowledge is in those relations. Okay, I made that.

That is. That is knowledge.
If I said tell me about USC, you'll continue little or tell me about schools in L.A. It'll list USC for you because I put that in.
Okay. Likewise. The knowledge here is labeling. I grew up bounding box around a chair and went to drop down and a picture.
I did that 10,000 times, said not a chair. Have like four legs. Okay.
It knows, meaning that the weight somehow. No, still doesn't actually. But then the third leg over there, nobody label anything.
Nobody do. I made any knowledge graph. Where's the knowledge?
Yeah, exactly. The knowledge is in the words. That's all the knowledge is in the word order.
Okay? The knowledge is literally in the works. So then I push back so strongly it's not even funny.
I push it so hard. You fall on the floor. I tell you, that's B.S. knowledge, because word ordering is not real knowledge.
Word ordering is word ordering. So what I start to really compute for you in a writer term paper is computing word ordering.
Okay. To you it looks like knowledge because you can make sense of it. It has no idea what the [INAUDIBLE] it did.
That's a very important philosophical thing I want you to know. Okay. We rambled on for so long.
Hey, let's take a break. We should do a prequel. 658 You guys.
A five minute break. A ten minute break. Sorry, that. Come back in 10 minutes.
All right. Hi, how are you?
Yep. Welcome back, ladies and gentlemen. All right.
I'm trying to be Elvis. Okay, But I cannot be Elvis. Hey, cool.
Check this out. So we. Are doing work in that, right.
I seriously will play all of you. I mean, sit your butt down in straight French.
Murder a civil. How.
I swore an oath in Chinese, but almost nobody noticed.
All right. Uh. Carol.
Oh. You know what's coming up next on my little list here?
Attendance. Oh. Oh, no. But, you know, let's actually get password net.
Okay. So check the Internet. Exactly like all of that. Right.
But then in a more text format with links, that's really all it is. Okay.
You see a vocalist, right? A singer. A singer is more general.
I mean, like synonyms. What else would you call a singer?
A vocalist vocalize her vocal research in a British English accent, and in some humans set it up.
Then if you say, Give me an example of vocalists, then it'll tell you, you know, Bailey, Johnny Cash, I'm whatever, right?
And you can say, Give me an example of French vocalists. Then it'll go through the list and make sure that they are French.
You know, You see how it works, right? So it's not magic.
And then word night is something very carefully constructed by hand, and then the system simply uses to influence it.
And so it does come up with obviously good answers, you know? That's all. But the common sense thing that I told you was more philosophical.
It is not able to talk about the world. You know, I'm just going to tell you one quick thing about psych.
Look what happened. So we would ask psych something like, you know, I'm going to Seattle.
Supposed to tell Psych, I'm going to settle in. Psych would say, make sure to get a car that has good rust cutting.
That is a factual thing. It's a good thing. And then you can query and say LaGuardia or LaGuardia is a chain of entrance.
It'll tell you you're moving to Seattle. It rains a lot.
In Seattle, rain makes steel rust.
Cars made of steel. Therefore, rain makes cars rust.
Humans drive cars. You're human.
Probably going to buy a car there or the car that you already have unless it has good rest proof coating your car with rust.
That is all nice. It clearly told you something that is useful.
But then if you ask a little bit more, why do humans care about rust?
No answer. Complete B.S. Okay. A human might tell you because it looks ugly.
Do you remember this? You know, we were like things like esthetics, right?

All There is no way in [INAUDIBLE].
So you cannot make a knowledge graph piece for the millions of things we use in the real world that we call common sense.
And then there is a lesson from Psycho. You have to actually live in the world to find out what common sense is.
So I am now a big critic of air in the word common sense reasoning.
The word sense means from sense organs. It means you have a nice piece of nose, your mouth.
Right. So if you substitute fake data for that, it's not the same as sensing it is second hand sensing.
That's why a robot that's running an alarm. So not in that. I don't want to go off on a tangent there.
But the idea is you can make the the word net and come up with useful hierarchies and you can create them.
Okay, so Wikipedia then a few more years go by afterward.
Net Jimmy Wales decides all of the encyclopedia knowledge held in the world
by either Microsoft or the British people with the Encyclopedia Britannica.
So before Wikipedia there are two massive. Knowledge repositories in the world.
One I'm going to Google them. One this a bunch of fat volumes that I call IB Encyclopedia Britannica.
It came with a big letter binding and they were pretty far right. But the joke is they've become obsolete the second they're printed.
Okay, because they're like dead words sitting in prison. So can you talk about Ukraine tomorrow?
The history of Ukraine is going to change. There's no way in [INAUDIBLE] they can update them in their timeline.
So now, yes, you have encyclopedia subscriptions or a Britannica subscription, but many people still think of them as these massive tomes.
You know, these books, I mean, look at this. So Jimmy Wales wanted to free that, meaning give it give access to the whole world.
So look at the second one as well. Microsoft, they had a thing called Encarta.
Look at this. Oh my God, does anybody here picture themselves actually picking up one of these and going to pay 187 and reading about the Philippines?
Okay, There's no way in [INAUDIBLE]. Just Google, dude. Okay.
So, you know, that's very sad that they still sell them, but they're considered like an authority, by the way.
But the world changes so fast that there's no way in heck all those people together that know like, look at this, 2300 bucks.
It is done by part by rich people, but it's not, you know, against the bare warlock of the words.
It's a show piece. Nobody instead of read them to read them. It's basically crap.
Okay. So likewise, you have Microsoft and Carter.
So Microsoft said, Hey, we can do this to C. So then they published a bunch of CDs or CD.
ROM is what we used to call them, and now suddenly they are like all electronic and everything.
Windows 95 Oh my God and card of 94 hours.
I left all this and I know and Carter kids, it's like a junior version of Bitcoin.
Okay, All right. Same idea. But as soon as they type all this up, you're becoming obsolete.
So Wikipedia is an attempt to say it is going to be as modern as tomorrow.
If you ask about the US elections in a couple of years ago, it'll tell you because somebody went in and made a page about the election results.
It's maintained by a small list of volunteers. Okay, what is it here?
It's made by it's maintained very carefully the authenticity that the integrity of that security by a list of volunteers,
they all basically police each other. Some.
Nazi person cannot just go in and then say Hitler was a nice guy.
You know, the Holocaust never happened. People, by the way, do exactly that during the Holocaust.
Page by Wikipedia. I'd say the entire thing is made up of Jews.
Wow. Right.
So then, thankfully, within a few seconds of some idiot doing that, the volunteers go and put the page back to restore back what used to be okay.
So it's not easy to basically put B.S. in there. It's a high quality human created in a knowledge base.
But alums, unfortunately, are actually poisoning Wikipedia.
That's very shocking, right? In other words, there are some unscrupulous editors that sign up to be a content creator on Wikipedia and radio.

Use an alarm to not purposely in a recent accident, come up with things that are not true and add that to high quality Wikipedia.
How sad is that? Right? It's poisoning the world, don't you agree? Salem's poison.
Poison Wikipedia. It's sad.
Launch language Model C on the person and not this.
Look, this one. So here Slate.com basically says, you know, yes, they're going to do it.
But at the same time, we know Wikipedia will survive. But if you're actually going read more of these, okay, I mean, look at this.
So people try to like rake like this. People try crazy things again.
And that's not good because for a while we can basically try to stem the tide.
We can push back against all this, but the volume of the auto generated crap is going to be so high that human volunteers would give up.
It's almost like being flooded, you know, like a little wall of water I can push back.
Ten of us can push back a lot of water When the wall of water is 10,000 feet high, just basically stand back and say, I'm going to die.
There's nothing you can do. Okay. It's really sad. So you should go look at it.
Yeah. I mean, there's so much data. Yeah, This is the whole idea of data poisoning.
Like all of that. Okay. Only one cell alarms.
You know, it shouldn't. Maybe the word poison is too strong, right?
So produce, like, bad Wikipedia pages. Yeah. Search for the right words.
You'll actually get it. All right, I want to move onto the Wikipedia Wikipedia value.
Yeah, I felt like that. That whole notion of verification, that is really the key.
How and what human or what automatic way can certify that something is absolutely not wrong.
Okay. This is probably what I was looking for. I is tearing Wikipedia apart.
Motherboard. You know, I started coming at a reputable site, right? So then that's why they you see like this.
You know, on the one hand, yes, such a beauty can summarize Wikipedia articles in the other hand.
That's poison. That's basically poisoning, by the way, because prompt attacks.
So prompt attacks will insert themselves between you on the prompt,
and then you might have a good prompt telling me about the Jews, the culture, the Jewish history.
The prompt might be poisoning poison into telling me about Nazis.
Exact opposite. And you will get something that you didn't want to look at, right?
So even though the Wikipedia retrieval is okay, but you can screw up the summarizing because by injecting like, bad things in there.
Right. It's very sad to see this citing papers that don't exist.
I told you. Okay. Fabricated. And on and on and on. Should you believe Wikipedia?
You know, in the US, by the way, many high school teachers absolutely forbid students using Wikipedia as an article source.
You cannot cite Wikipedia. Okay. Liberal, actually. So then what are we going to do?
It's going to become more and more hard, right? She added that someone checked the sourcing, but like I said, you can do ten, you can do 100,000.
How are you going to do a million? Okay. And it will scale up to that, sadly. So it's basically you cannot fight with the machine.
Ultimately it comes out of the right. Yeah. Okay. Anyway, so hallucinates in area.
Every link, by the way, is very cool. By the way, in the European Union, this happened last week.
Also suddenly we have the air love.
All the EU countries got together and said we'll make a special GDP rule for eight where if any citizen is harmed by air, they should be able to file a report. Any air generated content should be explicitly labeled as created, otherwise the company is going to get fined.
On and on and on. It's actually very cool. European Union.
The US has no law like that yet, but it won't tell you so.
Meanwhile, all that aside, Wikipedia has a very cool idea where anybody has access to all the knowledge in the world, because in Carta and Britannica it costs a ton of money.
You know, you're in the news. They used to eBay 2300 bucks, right?

So then this guy, he's a cool guy, said, I want to actually change all of them.
See this 100 gigabytes? Okay. It's crazy compressed and then uncompressed is terabytes.
51 million pages and counting. So neat.
And it's in so many languages in the world, by the way,
people are actually using headlamps to translate Wikipedia pages into some of the lesser known languages in the world.
That's actually a good thing. So it's like an Ethiopian language.
Many people relatively don't speak in English, so say there's no Wikipedia.
In Italian language, you can have a language translator and like a Bert Transformer to actually translate from one well-known language
like English to all the less known languages so that kids in those languages can learn in their own native tongue.
That is useful. All right. So I mean, at so it started back then at the turn and one took off when it.
Nice. Got there. So Wikipedia and you know all of this, right?
Neutral point of view. This is extremely important. You cannot have any editorializing.
You're not allowed to say Israel sucks. You're not allowed to say Palestinians just cannot go straight to the facts.
Okay. Just. Just the facts, ma'am. No spin, okay.
It's called no rain, no slant. It's just the facts. And then anybody can edit.
Okay. But then, like I said, if you do look where things. As soon as she added, the people that maintain it officially,
they all get notified up here at it that they all read it and they're basically approached to approach it silently by doing nothing.
If it was actually good. But if you purposely, you know, if you're up there looking over it.
Okay. Yeah. And there's nothing is copyrighted. Everything comes with the copyright free sign.
You want a diagram? You all have what is called CC license.
Okay. Like a Creative Commons license. It means you can freely share.
All right. So be civil. That's very simple. And yeah, the first rule is there's no firm rule.
Okay? This has worked for the past, like, 23 years.
Okay. Look, I need several. Okay.
And then I will ask you for all of these. Yeah. Okay.
So many people come into Wikipedia because when you search for something, Google will see if there's a page for Wikipedia and make a link.
Copied it up because Wikipedia is a trusted source, so they get many visitors through Google AdWords.
Nobody goes to Wikipedia and searches for the Philippines. Instead, I go on Google and search search for the Philippines,
and then the Wikipedia page would be at the very top of the link result because it's good.
And then, yeah, so court rulings actually cite Wikipedia because many US Supreme Court cases,
there's an article for each one of them written by actual experts. Okay. Hey, on that note, Oh, my God, LexisNexis.
And I tell you this LexisNexis. So very important information source Information resource in the US.
LexisNexis. They used to be in Columbus, Ohio.
I went to Ohio State out by the building all the time from the eighties when there were high
quality medical and legal information only so doctors and legal lawyers would go on strike.
Recently, they started using tragedy to summarize some of the amazing lacrosse players that they had.
OC and L am screwed up.
So they basically hurriedly tucked their tail between their legs, apologized and removed their reasoning summarization claptrap.
If you want, you can just tell them. I'm not sure if it'll come up, but I am problems right now.
Other words, you know, it is not ready for primetime at all. Right. Okay.
Scientists generally around limited. Okay. Select this right error message then that section, the number.
Yeah. So the idea would be that some lawyer would have a high quality summarization of a thousand page,
you know, like a legal argument back and forth between two parties. Tell me in one paragraph what is is about.
It'll be cool if I can do it for you, but you might not be able to trust what it tells you.
And that's worse than you know, having asked it in the first place. Write the words like, Yeah.
And also, you know, I mean, this look very bad and going on this. Okay.
So summarization is not easy. Okay, So Wikipedia founders and all of that, right?
Steve Jobs, you can skip all this. Yeah.

This is an example of named entity recognition better than a page of what Steve Jobs.
Right. Has so many things. Steve founded Apple and he's also in the Disney board of directors.
He basically purchased Pixar, you know, I mean, he also founded Apple with this other guy to the board, become co-founders.
So you can see in one page how this whole idea of knowledge graph works.
Okay, every link is wired to another page. Okay.
That's a ball of all of them together, which is called DB PDA.
That database PDA, it's called Wikimedia Graph here and.
All right. So again, you see categories say basically link that one Steve Jobs page to all of these.
That's where the ontology comes in. Okay. Because oh, by the way, he also briefly had a company called Next.
I used to work on the next computer. Okay. It's beautiful. This amazing.
Almost like a newspaper type in a big computer. But anyway. All right, so there's no taxonomy.
Yeah. This one ultimately is a ball is a graph of any node.
Cannot do it anymore based on like the Internet. Okay. On the Internet, nobody can say, please be a hierarchy, okay.
To how anybody can link to anything can Wikipedia.
It is not a hierarchy at all. Any link can go to any link.
Okay. But still everything is a triple. It's three pieces, just a little three piece subject predicate object.
Okay. And then you this I'm going to skip, right? It tells you, right?
If you go and look at any place you can go and do all this basically tells you like
all the different things you can do and all the other things except from other words.
Every page is carefully constructed. This all okay, but then it's all about links, you guys.
There's actual links, meaning this object links like one entity or other entity is also category links.
You're asked about Elvis Presley. You can find out about blues singers. I'll go to a higher note and give you more of the words.
So your search for this, what is saying is I'll give you a here so that you can also search for all of these that are siblings you knew about.
Elvis put it in a number, Clapton, you know, or somebody else.
So then you can discover all of them by going up and then going down like that, too.
That's all like possible to the make lots available right there, if you like.
Know where to look. In 3 minutes, something will happen.
Something's going to happen. All right, So then Wikipedia and whatnot.
Yeah. You know, why the heck not, right? Because you can actually you can imagine, like, a world where those two are related, right?
Because world net is basically all this categorization. And Wikipedia is actual objects all wired together.
So that can be a correspondence between them.
Meaning if the word and it says corgi a type of dog, and it can go to corgi in Wikipedia and then find out about Corgis.
And then it says, Click here for more about dogs, click on dogs.
And all the other dog breeds appear exactly like the word in it. So Wikipedia makes words that basically become true.
Yeah. Cool. And then Wikimedia Foundation.
Yeah. So this is actual corporation that is nonprofit that is making Wikipedia happen.
This corporation isn't troubled by the way, You know, it's been funded by volunteers.
And now basically wealthy philanthropists are like knowledge free access to knowledge.
But gradually the money is running out. And so, you know, they're basically begging for money.
If you go to Wikipedia, hey, please don't click quick and send me five bucks, you know?
It's too bad that they have to beg like that.
Even archive.org, which archives billions and billions and billions of pages by a completely different billionaire philanthropist,
also ran out of money. So they're also saying unless we have the money coming in will not exist anymore.
And that's very sad, right? Because then on the Internet, things will permanently be gone.
Okay. Hopefully it doesn't happen. Right. So then you have a dictionary and you have an own with your data.
Yeah. So we could data is actually what powers the whole Wikipedia pages.
In other words, the actual graph on the on the back of it. So we get it.
Okay. From graphical PDA. I think I'm going to tell you that next. Yes, exactly.

So you have the Wikipedia data. How do you make it into an actual like a knowledge graph?
Right. That's some in transition can search for. That project is called wiki data project side aims to create that so that it means you can then
take the Wikipedia articles that people write and go to every link and make a triple subject.
Pretty good object subject. Pretty good object.
So one page can give me 100 triples because one page has 100 links to 100 other things, each link becomes a triple.
I'm the subject I'm linking to something. So subject, predicate, link, object, whatever I'm linking to.
Amazing, right? So then that idea has got to be key data to wiki data and then becomes this triple triple version of Wikipedia.
We keep data cool. Okay. You can even by the way, I Google all this.
Okay. It has the same information as Wikipedia, but not in a page form.
More like a triple form, you know. Because it is rebels who.
Oh, all right. Same idea, okay?
It's all about, again, these hierarchies. So look at this examples that I've been getting, right?
Just about San Francisco. It's a city in California, right?
California has a city in the United States. Actually, it's a state, an interstate and then San Francisco, a population.
Mayor That's all. Now it's like London, right? Yeah.
And then geolocation that long, the one change quote or average number of visitors,
you know, is simply tax that you can put in and then reason over the meaning of them.
And you have multiple languages and are told about all that. These days you can increase more languages by Al-Alam.
Alabama can help translate. Okay.
And then here to compare Wikipedia versus they're the same. I can send Wikipedia that is actual page that you wanna go and read and wiki data.
It is more structured into subject object, predicate, but exact same information back and forth.
You can think of this as the background and this is the actual text that goes along with it.
So what if you remove all that spurious extraneous text and throw it away?
You can simply do Douglas Adams link to every one of them. That's what all of these are the words.
That is what's underneath the page that you read. Okay, cool. So you can see how many nodes you can create.
So to finish all this up, Google has a knowledge graph that is in part some.
Some of it is their own meaning, their index. So many documents, right, to create all this embedded index.
They can go in their documents and do named entity recognition and build their own knowledge graph.
And also they are allowed to use Wikipedia as knowledge graph.
So they have some crazy, you know, amalgamation of all these knowledge graphs that they use to basically answer your question.
My question every day this has been quietly going on.
2012 Hummingbird Update The site I showed you last class search engine land or com.
It has so many things you can go and read about Hummingbird. Yeah.
That came from a project called Freebase. I actually completely forgot about that. There was an open source project called Freebase.
Okay. Okay. So if you don't know the drug paraphernalia hasn't something called freebasing cocaine, please don't overdo it already.
But anyway, so that's why it's a joke on the word freebase. Okay. So freebase, then?
Freebase made this thing called knowledge graph, which basically Google got right.
But this actually won't happen. People said, you know, you're trying to compete with Wikipedia.
Jimmy Wilson, his follower, said, We already have an open source project already.
You have volunteers and philanthropies giving us money and we should all put our efforts into making it even bigger, not try and compete and go off and do something on your own. Okay. But Google, like everybody else is after making money, right?
So they wanted their own that nobody could ever take away from two. Then that is called knowledge graph.
It's weird. You name it. Knowledge graph, right? It's a generic term. Okay. All right.
So then, yeah, that's all. And see how they make not this universe actually.
So how do you power knowledge graph meaning where do the entities come from? They actually come partly from Wikipedia.

So go to Wikipedia. I take their stuff and make it your own and I charge people money for it.
Right? That's so bad. That is what opening I also did when the NY Times took all the articles and started charging people money for charity.
So NY Times sued them and said, If you keep doing this, we're going to find you cool.
So then they get LinkedIn and LinkedIn, all these trusted news sources in the world you have to have trusted in your sources and then you can take it,
send things, not strings, you know, In other words, those those abstract wires.
Right. You understand they don't mean anything because they connect real world things.
You know, people, places and entities are real when we search for them.
We don't. Not because we like to play with the graph data structure. Okay?
Most people don't even know what that is. So forget the geeky stuff, forget the techie stuff.
Focus on what is underneath that, which is content being linked together. And then you can watch the thing if you want.
Okay. Get ready.
Oh, my. Oh. Lacy, She.
Cool. We start off on a good note?
One out of one people, Tonka. Cool.
Two out of two. I like this. Okay. Imagine in such an influence.
This perfect Slovakia record is very good. 100%.
Nikita. Oh, no, no.
It's two out of three. 2%, three cold or now it's going down 100%.
66%. 66.66%. Oh, no.
So there's no Nikita. Okay. Scotty or Scotty.
Or Skokie? I don't know. Oh, my God.
Now I wish to order for Doctor. 50% or no so bad.
Dharmic. Wow. Okay, so three out of five.
Although one book. Karim.
Wow. Four of the six. Really good. All right.
How is that possible? I screwed up on the map. Well, you know, I'll go back on the radio.
Okay. So who is missing? Still missing just now.
Okay. Ratio doesn't look at GPA. Okay, So I had to make it go back up.
Be careful. Okay, So far. Now, that's enough. Knowledge graph accelerators.
Where does Google do? Right. They will look at your query term. All right.
They have all these word equivalencies made by other people. Taj Mahal is a modern like a music band, right?
Some real time concepts. But that's only because Taj Mahal was a thing for many centuries.
It's a tomb that the king built for the queen called a mausoleum.
And so you're searching for Taj Mahal. Will either look for words about mausoleum or look for the musician Taj Mahal.
You have to disambiguate. Okay, So the human knows what they want.
If you want the Taj Mahal musician, you can say Taj Mahal music and then it won't show you the actual building again.
All right. So, yeah, deeper and broader results.
So this what I showed you earlier, right?
When I search like Picasso, they summarize what a knowledge graph has into a little like a card, almost like a baseball card.
Okay. That's very neat. And they also call it a snippet.
It's quite a rich snippet. So snippets, sometimes they're not for us. Okay.
But no, you know where the snippets are coming from. That came from a knowledge graph.
There was no pre done card like that that's sitting there waiting for you to Google it.
Okay, great. And then they'll also show you like Matt Groening.
So what about his daddy? No, no, no. So I went to see Kung Fu Panda four yesterday.
And obviously, Jack Black is it? I know Jack Black's birth mom and adopted dad.
So funny, right? And it's such a crazy, cool thing because actually that artichoke blood but source po.
So for both dads up the ghost there and also the actual and apparently dead.
So funny. But it's actually this real world component to it.
Jack Black has adopted. All right, then.

Yeah. What else does it do? It provides a summary.
See that summary? Okay. This is where things like Jeopardy is going to take more and more roles already.
Prajapati, They use something called Bert a Transformer to make the thing about Tesla that you see on the right.
But now they can use Gemini, which is a much more expanded version or an opening I can do.
So summarization will take the place of raw links more and more.
What if this becomes center stage at the bottom somewhere that tell you, Oh,
by the way, here are the links from which some of these came, I think very slowly.
Research is headed okay and hopefully they will never remove the fact that they put all this here on the bottom.
If the actual links to the sources are gone, it is like writing a research paper with no references at all.
The world has no idea what part of what you said is yours and what part your boring from the paper.
So does that make sense to you? That's like every research project must reference things.
Hopefully our search summarization must also reference actual URLs.
Okay. How many of you know about the new search engine?
You know what? You right. You know, where you you a search engine.
You're always actually pretty cool in you. That's actually what I'll do.
They ask about something, right? And then it'll give you a summary, but also list every single source for you, but sort of Gemini for a while.
Try to bring it in and out of that. Okay. But no, they actually do it anyway. But I hope that never changes.
Okay. Because it'll be actually very sad away from you.
So again, we went off on like all sorts of tangents, but it's actually useful a tiny bit more here.
Yeah, last few. Okay, so knowledge graph again she look at this.
One more example of knowledge graph. Okay. There's so much. Taylor Swift and then Joe Alwyn.
Oh in the past And then further retro.
So every person has all kinds of other people relate to them.
Every person knows also the place where they were born, some objects their own, maybe like what kind of car to the driver and then his partner.
And then she has. She lives in the own world. Right? But now the two worlds collide because our partners.
But now you can change that the locality in our call. So that is how it works.
Like everybody is like a cool little node with a whole bunch of links and then so are you.
But now we can merge and we can be super graphic. It's an easy, obvious idea.
It's a great idea. So our last slide literally says this.
These days, when you search for something, it is not exclusively the inverted index.
For the longest time, this was the only source where you typed some words to give you all the words that matched the idea for this news article.
But now, increasingly, they also use knowledge graphs because they look at what you type.
What named entities can they extract and what you type and go going look in the
knowledge graph for those notes because some system already put nodes for those names.
In other words, I type Bill Gates and I'll do it again. I mentioned his name a lot.
Well, let's type Elaine. Why not? Okay.
So as soon as you do that, you see all those pictures, you know, And then what about this part?
This all coming from summarization. Okay, so let's call this number one a Wikipedia article about it and then all the top stories.
It's all wonderful. So the idea would be that, yes, the Google in real things, in real time as we speak one day ago,
but also going to a graph and then showing you because these facts never change.
Right? I mean, how would you change it? So therefore, you can use both, which is exactly what they do.
Wow. Any questions on this? Next, I'm going to go to pressure and pressure and thankfully is a very simple idea.
But even before I tell you what President President Bush is great president is what started Google.
Google had the whole founders, Sergey Brin, know Larry had one amazing idea,
this little algorithm that can be described in a few lines, it can only verbally tell you right now.
And they went to their advisor and said, Should we start a company? Yeah, go for it.
Well, quote. So then even today, when you search for something, the order in which you get results back is partly driven by this idea called PageRank,
where they rank the pages that came back from the retrieval.

In what order should I say with the page? Okay. So they use one very important metric, that metric called PageRank.
The higher the metric, the higher that number, the higher the value of PageRank. That's what I mean.
You do a word search, okay? It comes back, say comes over a hundred documents.
In what order should they give you? The 100 documents for each of 100 documents.
Compare one floating point number called a pagerank, 100 page rank values.
Okay. I sort them numerically and take the top ten PageRank values going from top to slightly from big this month and put them in the top page.
One, two, three, four, five. So the very first top page rank is literally my rank.
Number one result number one, two, three, four. It is that simple.
But increasingly that is not the only thing they use.
So page rank might come up with some ranking,
but the ranking could be modified by the fact that the president came up with a Wikipedia article in rank number four.
But Wikipedia has a very important source, right? So this new thing will move Wikipedia to the top.
So page rank is not the only source of truth is what I'm trying to do.
And increasingly Al-Alam, believe it or not. In fact, L'oms using RAG can be used in any search engine to actually re rank.
To make it a better ranking, and then we all want the best ranking possible.
And then what we want is you type something. Wow. The very first thing or the first thing that came up was exactly what I was looking for.
You saved me so much. Time to look through all of them. That's what you want.
So ranking is the idea that you already have a ranked algorithm came up with.
Can you possibly do better? Pharrell Williams can actually help you do that.
Okay, all that aside, I'm going to tell you now about ranking. Okay?
It's called it's one of the easiest algorithms. And I even have a sham magic shell thing you can actually play with to learn about it.
So we'll see. Okay. You can yell out some questions if you want.
Here. Cool. Here's your patient and quirks.
Okay. Supposing there are five of you.
We're gonna pick ten of you. Ten of yoga and ten of you want to start like a club together.
Okay. And then the immediate question is going to be, who's going to be the leader?
Like a boss. Okay, Tell the other nine what to do. Okay, so then all of them enter into a little voting scheme.
Secret vote can give you all a piece of paper.
I mean, you can write down everybody of the ten of you can write down, be a top three or four people that you like in some ranked order.
Okay. You can watch this recording.
So each of the ten can vote for only one person they can support for themselves, or they can vote for three other people.
Likewise, the Australian people also vote for the great idea. So then I take all the votes and here's what I do for everybody over the ten.
Everybody I count. Who else wanted you?
Like all incoming links. Sam, CNN dot com.
How many other pages in the world sites in and out come as the news sources?
Many million sites. Okay. So I had all this numbers up. That's a pretty high vote that I'm getting that makes them popular.
Okay. That is one number that I get right. I would actually then divide that by me how many other people that I vote for.
Because if you all want me right, and I in turn want many of you, I'm throwing away my own importance.
I'm diluting my own influence. Right. So there has to be like a negative. It has to look away things down.
It goes in the denominator. That kind of a measure is called a pagerank.
So we do that for all the people. Okay. How many people voted for you as opposed to how many people you work for?
That's a number. It becomes a floating point number. You have to iterate, by the way.
So I'm sure that when the iteration settles, all the ten of us have a single number, standard positive decimal number called a pagerank,
and the highest PageRank becomes a leader in the search engine that becomes the first result.
Then second place ranked third president. What makes sense, right? Wow.

So easy. Okay, so again.
Yeah. So then supposing I'm talking about me, right? If you are, you wish Rush only votes for me, right?
He's concentrating 100% of the votes for me, and I'm thankful for that. But if he also votes for four other people, my vote is diluted, Right?
I'm only one fourth important because you pick three others. Okay. That's what the test.
So praise rank is that very cool idea of doing that iteratively. Okay to say who voted for whom.
But then let's find out the final result. Okay, so let's go. Where Page Rank came from is research papers.
Okay? I told you in a research paper you have to actually site city site.
That means reference other papers.
Otherwise many conferences will completely reject a paper because almost nothing that anybody in the world comes up with is entirely their own.
You have to have referred to something else. Stand on the shoulders of giants. Right?
So that is called a citation analysis. In other words, citation means you analyze how many references did you refer to and those references,
how many other references that referring to them like this. Okay, so pushing this your paper in a paper.
I know you referred to all of these references support this reference here.
It was also referred by so many other references.
Okay. That becomes a very influential paper because they're all in other words, in this paper, this reference point in the paper.
It's so important that just like I referred to it,
I'm going to draw the arrow outwards my paper to a reference for paper to a reference Pepito reference.
That's a very important paper that I cite, right? Yeah. So this paper is not that important.
Only three areas have pointed to them.
So therefore I can then find out, you know, given like a graph like this, what paper's more important than what all the other papers.
So it was after many years. My paper here right.
Is being referenced by 10 million people and suddenly my page rank in this graph could go up much more.
Because, you know, I started by referring to somebody with a pretty high appraisal and I became more and more important over time.
So these things can evolve over time. Obviously the whole ranking stuff, but you have to start somewhere.
So Google assigns in all ranks before they give you the result in the words so the order,
but they're also rearranging knowledge graphs and increasingly they will reorder using rags.
Actually, you know, if you remember I talked about here, I so I'll just simply Google it and you can read about it.
I just said cohere l am re ranking.
See that. That. Circle here.
There's also an air company down here. So they have a very simple I mean, you can do it right there.
Okay. So-called like this, what is it, capital of the US?
And then it is using rag with all these snipers that came back from a knowledge graph somewhere to answer the question.
Okay, you got all this right. And so then see that that is the ranking literally easy, high, medium, very small, very small.
But then using rag, you can say, you know, are you sure it is very high?
Can you rank them? Possibly. And if the ranking produced this is a new and higher value that will actually go to the top.
Okay. Okay. Select this. So they have a whole API for that.
You can go and read this if you want. Okay. Like this you can now call.
Well, by the way, you can also rank documents based on your own need.
Suppose you are a law firm.
You can then take a whole bunch of PDFs about lawsuits and use it as an extra training step and make it being the rank for that.
The standard search engine within your law firm produces like a bunch of results.
Then you can use this extra step to rank and get hopefully a better quality ranking.
All right. I guess I shouldn't basically jump the gun to go ahead and tell you what president because all idea of ranking.
Okay, That's all. So Google has this algorithm and I'm going to tell you what that is.
And yeah, the importance is slowly going down over time.

There's an alternative called the hits algorithm, purely for why I wasn't asking the exam, but you can Google it, though not the first actually.

Larian in our search to come up with the ranking idea claim that came before that claim but did not start Google.

Oh! Oh, my God. What happened? Wow.

Okay. So the notion of link analysis. So the idea is again, to point talk about links.

Okay. Oh. Something happened.

Share. Sorry.

I have no idea when I clicked. Okay. All right.

So the idea was just telling you that our link analysis or the link analysis, they call it Biblio metrics.

Biblio means books. Metrics means measure to measure books in relation to each other, and what books are commonly referred to by their books.

Okay. If I write a book and many other people refer to my book, my book becomes important.

It's high in page rank. Okay, that's all.

So then, given a bunch of books that are linked to each other, the question is what is the most valuable book?

And you should read that first PageRank algorithm. That page should be sold at the very top of the rankings should be a result number one.

Okay. There's lots of things I can skip here. Okay.

That's somewhat beside the point. Okay, So again, frequency patterns, graphs of citations about citations, which is what we're going to get to.

So basically the Google idea was we can treat the whole Web as a research paper situation where I make a web page, you know, satellite estimate in there are linked to all kinds of call Wikipedia, attention, geography and Smithsonian.

Then somebody else is also doing the same thing, but somebody links to me, you know, the link to satellite stereo.

So after a while, we get this graph of random links to each other, right?

So given a graph like that, if somebody does a search on the whole graph, what page should be searched first to that person?

Try something. And their entire thing is based only on links.

How many links come in to you? How many links got out from you? It's all about in and out.

Okay, That's all. So obviously this is a problem because not all pages are the same.

Some pages are, for example, the extreme right wing, horrible evil sites.

On the other hand, you have National Geographic, BBC, you know, New York Times, Time magazine.

So I would only do link analysis. You should make some links.

Way more than that is whole ranking idea.

Therefore, the you start today at Google that says not all pure page ranking, but the very first iteration of Google was 100% page ranking.

All right. Okay. So let's talk about again, this notion of similarity.

You know, see? So two documents become coupled and we become coupled.

When they cite the same references. These papers already exist.

I wrote a paper called A and I am pointing to this, this and this.

You wrote a paper, Colby. You also point to the same references.

So you and I become linked somehow because you are pointing to the same source again.

Although you and I obviously don't link to each other. We could, if you want.

But even without it, we are pointing to the same. Or drawing drink on the same.

Well, okay. To become friends. Quote like literally over here.

Okay. I'll say side c h h d a sites e as well.

So these are c, d, b also sites said this quite common thing.

You know, if you go to archive. I want I want to search for things but just to tell you if I go to archive she has papers right.

Is incredible how many papers are published every single day.

Crazy. See that new. Well.

All of this recent data collection is dead, right? So many.

So if you click on any one of them. Then immediately you will see a whole bunch of other papers being cited.

So archive is a big ball of all kinds of paper citations, right? And so on.

Google Scholar does the same thing. You know, just from an academic ideal, this is great.

They're all like very similar publish metric, you know, things that are being in the world.
Okay, so then this notion of a journal impact factor. So, again, you know,
some journals become very important when they're cited by when articles published in those journals are cited by lots of scientists.
Like Nature magazine or Science magazine, you know, or any one of the air conferences.
And I clear to play in Europe's in order room because they're all high quality journals and conference proceedings.
So they have a high impact factor, meaning if you publish a paper in one of the journals, then you will actually probably get promoted.
On the other hand, the exact opposite of that. What do you think it is called?
That's a word for it in the world. So it is neutral.
Okay. It means if you publish a paper there, nobody would care.
Not a bad thing. But here are the high impact journals where it is highly worth publishing.
The whole world takes notice. But what is on the side? Very bad journalist, but there's a word for it.
What is it? Yeah, exactly.
Because a predatory, predatory predator means you prey on things.
So predatory, general. So bad. Absolutely horribly bad, because you can submit crap to them if you want to publish it happily.
And they'll charge you money, you know, $200 per page.
Okay. Apparels is yucky. People are done experiments where they publish 100% bias.
Okay? And then they publish it. The great sad, shameful.
But sadly, there are universities where there are people that work in universities.
They publish or perish who they don't publish, they'll get fired.
So the good quality journalists are rejecting my paper because my paper basically sucks.
But then who will publish them anyway? Just the opposite of that.
All right, so the Impact Factor Journal, you know, so I'm not going to read all this for you.
Okay? So I'm just going to skip it because I want to get to the main part page rank now.
So different journals, all these impact factors, you know, so these are things that are household terms.
Okay. If you're in communication, so as in tutorials, right.
In the business journal. So the lower the rank, you know, is not highly with publishing.
It doesn't have too much impact. Okay. Okay. Now, the citation graph, this web page rank starts.
Okay, So say you have a paper, you can cite some of the other people like an Instagram.
I can follow a whole bunch of people that does not give me a high praise rank at all.
Instead, you have so many people follow me.
These are all outgoing links. When you when you follow somebody, it's called outgoing link.
Okay. That's not a big deal. What is more important is how many followers do you have?
Other people are following you, right? That is called incoming links. So the higher that, the more impactful you are.
Safe. Again, some kind of a maybe like a cosmetics company wants to run some kind of ad
campaign will obviously go to people that have like 1 million followers and say,
you want to do an ad for us because I know that 1 million people are watching them, right?
Yeah. So says all of our incoming links.
Remember that it's in a web page document Are URL that is linked by so many other documents would go by and bridge over other words.
You cannot get higher PageRank by making over page and linking to every single thing in the internet.
Okay, that's too easy. And that's not how it works at all. Alright, cool.
And so site. Yeah, exactly. So this is actually very interesting, right?
In a reference there's no navigation. Reference is simply a static thing.
Although if you go to modern journals let's bring our journal like all of the front to make
many papers be online with reference you can click on it will take you to that paper slowly.
This is papers online are becoming more like web pages. Okay, so take this with a grain of salt.
This difference is they're becoming almost the same. But in any case though, like this in degree part.
So, you know, so supposing there's some you write a paper, are they supposed to show a paper already in the world?
The paper is referenced by so many other papers. You know, all of them are offering to me.
Right. That is because there's a high quality good paper, not because it is a portal like yahoo.com, you know, or CNN.

And that is a big difference between the web and research papers in the web.
This could be like BuzzFeed, okay? It could be like TechCrunch or something.
And so many other people link to the articles. So that is one reason why this could have a high praise you're putting in in a paper situation.
Research paper better be good, otherwise you're not going to get linked.
So that's a big difference. Okay, papers are not on printers.
Okay. And then, yeah, this is also true in a research paper.
Even though you have an arch enemy and academic arch enemy doing the same research that you do on high temperature superconductors, for example,
and you go to the same conference, you publish the same kind of paper, your grad student, you know, you still need to refer to her paper.
You cannot say to your conference people, Sorry, she's my enemy.
I don't care about the papers. Right? That is not your choice. Okay, You have to quote.
Right. But whereas in the web, that is exactly the opposite. You've got a Domino's dot com, right?
Yeah. Find a link to Pizza Hut dot for me. Right. Good luck with that. Obviously, that never even talk about the competition.
Right. So that is also not the same when it comes to citations versus, you know, represent different.
All right. So one more thing is citations are actually very clean because hopefully this review of them number to us,
a little joke sort of user like primary review,
a secondary and tertiary interviewer is the job is to read the paper very carefully and know what is missing.
Why didn't you cite this or this paragraph is wrong, right?
So citations are very clean, whereas on the web this is whole search engine game you play where you make bogus links.
You know, actually, I forgot to show that you write somehow. I forgot that. How spam.
Yeah. This one very briefly, I want to tell you about this. Right. See this spam policy?
See somebody bias a domain that has been expired. So there's some high quality medical domain that used to exist.
Okay. Medical devices dot com. For some reason that domain expert some spammer hacker is going to buy the domain.
Start putting spam crap articles, but the domain still says medical devices dot com.
So sadly people would mistake that on the Google medical devices they'll be so spam right that this kind of abuse actually it's called domain abuse.
So Google says they're going to look for that and basically because people are like kill content.
Okay. You make patients these days using alarms purely just to go up and search rankings.
So citations will not do any of these. Okay. I'll list our reputation again.
CNN dot com is very cool.
I make a spam page, but every other link is a CNN dot com articles because I want to become associated with CNN doing damn crap through the links.
Right. Google is going to look for that. Okay. It is so neat.
Please read the spam policies page because they give you so many neat examples of all these things that I'm talking about.
Silicon, this link here. Again, I'm not going to go through all this, but it's neato.
So what I'm saying is bridge rank is more dicey, meaning it is not as cut and dry, well-defined as citation analysis,
even though it came from the world of citation research papers,
it's more loose so people can abuse them and they do abuse them and Google tries to fight back.
So here's the page rank algorithm. It's all about weblink analysis in versus out the two of them.
Okay. And then coauthored by this, This guy is an amazing scientist.
Okay? Please read everything that he wrote. He's so quiet, makes no noise in the world.
You know, it's not like Yeah, and like and, you know, like you're like, okay.
But then his circle, he made a lab called HCI, a Stanford human Computer Interaction.
HCI Stanford lab. I think Larry and Sergey were students.
And they'd say, I went to the lab. I lived in Palo Alto.
Psych research for me was done the Palo Alto Services.
TEMPLETON So crazy that because of what I told you,
it's a vote where you ask all the rest to work for you and find out how many people basically voted for you.

And then you have to then subtract the influence of you voting for other people.
So what remains is the president. I'm going to give you many, many, many examples.
All right. So a link is a vote. Okay, cool. So no link means no.
What? Like I told you, there are ten people you're going to pick a leader. Each person does not have to rank all the other nine.
Somebody might only pick two of them and say just one. Either one of them. Somebody else might only pick one person.
You know, just pick the person. Okay, then then that person's president is going to go up, for example.
All right, So then president, because nothing at all about exactly what the content is, how many, you know, words on the page.
Nothing, right? Nothing at all. It is all purely just there on the graph.
Okay. So it is. All right. So ultimately, it's so interesting, right?
A separate institution.
You know, you have a bunch of pages, right, with links, having links into some very light links and others if you close your eyes.
Okay. And then you click on some link and you open your eyes, then that link, I guess.
Suppose you have a page linking to a bunch of pages and said that pages are also
linking to a bunch of other pages and this pages linking to some pages in here.
Okay, then if I blindly pick some page, some page and blindly pick one more page from that.
And if I do that many, many times on the Internet, it's not just this 1 billion pages.
Then if many people do this starting from many different starting points, where will they all end up?
And the answer is they'll all end up by end up at a site being pointed by so many other pages.
And you see why, right? Because randomly, I could have gone in here and I'm going to end up here randomly.
You could have gone in here. You could end up here. So the idea is, you know, randomly, even if people start at random locations,
the page that is linked to quite heavily by many others would be the one that people all end up on the average.
And you could obviously go somewhere else. Okay.
Supposing somebody you know in here and they pick that page and say the page is a dead end, meaning it's not linked to anybody else.
Yes. You're not going to go that page. So most people will go to that heavily linked page, but it is obviously can go somewhere else as well.
So it's like a probability distribution, not 100% of the most linked,
but a high number for the most linked, a lower number for the ones that are less linked.
Okay. Okay. So they made they got a patent for it.
Oh, interesting. Right? Yep. Solaris and went to Stanford University in 1980.
This all classic history, by the way. You wouldn't know that in the US patents are valid for 17 years.
So I think 14 to 17 I forget. But anyway, there's an expiration date after the patent expires.
The patent is in public domain. Anything that they invent and claim and made money, now it belongs to all of us.
We can also make money too. Cannot see you. So now that's an open outcome.
We'll look at all of us and you can read this if you want for fun. Okay. All right, so there's the president.
Oh, cool. Wow. See? Look at this.
Okay, so you see this?
This means it's currently me. So say you mean city okay and so vs anybody else Immigration legal decisions okay Gloria you're trying to be kind of an.
Okay then my praise rank is simply a summation of.
All the pages that are linking to me. Okay. One at a time.
For anyone, if you're linking to me. How many?
Okay, So first of all, linked to me, that is. There's some value here for me already.
Okay, So to start all of this, suppose there are ten pages we don't know what is link to what, right?
So on the average, we'll start with a uniform page rank for everybody 1/10 when there are any pages and a group of links together.
He starting president would be one divided by an equal, but will now do an iteration algorithm that additional with them.
After we stop iterating. It won't be equal. Will not be exactly one over.

And some would be higher, some would be low. But together at one.
Okay. Because initially one over in the third of the one.
Right. Say there are four pages like this and a link like this doesn't matter what the link is.
Initially we say that each page rank is the same for all the pages.
There are four of them, so every page rank is one for when you sum them, obviously it's going to add up to one.
But after a while what might happen is some of the page ranks might actually go up,
this bridge rank might become much less, then this might become 0.01.
Okay. And then this might become like .33 and this might become .33.
This might become .33. That is .99 plus .01.
This one radically changed again. The reason is this one word for like all three of them, right?
But none of them voted for it. Stuff like that. Okay. So the iteration is going to change ultimately what ranks will end up with.
But you have to start somewhere. So let's start with an even one over. And again, that is what this is.
So initially I'll take the one over and ranking that some page called will give me, but divide that by how many pages total that page ordered for.
If you only ordered for me. Right. That'll be one but average.
What's for like five people. That's going to be five. So I'll then get one over five.
Does that make sense. Yeah. So initially the primary on the right hand side is the same number for all the sites.
Okay. Okay. So one time then for every site, I'll take the uniform number that everybody give me divided by how they diluted my vote.
So after you do this one round, all the ten of us, for instance, will now have a different value of three.
That is not one. Do it every ten anymore. Now we do it again.
Take two, but not anymore. And then divide that.
This number changes by that and then do a second version of like iteration number two.
So all the ten sites will receive like an update. All the numbers do it a third time or the ten sorts of numbers that are big.
So keep on updating. After a while you will not need to update because the numbers won't change.
Yes. I don't know.
Is there like. Like, say, like someone.
But some guys would have. A list of what sites. I.
Yeah. There are link forms, sadly. Also click forms in other ways similar.
Yes. They usually in like countries like India, Philippines, Vietnam, you know.
But we will go and do this for a living. Yeah, they are.
I mean, that is all the notion of search engine optimization or this whole is SEO game playing where you do what you have to do with links again,
to see the hope that Google will then elevate your page.
But Google is also very smart in all these tricks, right? So it's an endless cat and mouse game besides have been playing for the past 30 years.
Almost sort of simplest ideas like that are not useful anymore.
You Google discourse a page with completely empty crap. All it is is links.
Okay, then the one that actually they are actually down. We still punish the page, rank the presidential even more.
Okay. Yeah. But anyway, I want to show you examples of this.
Okay. Okay, so like I told you, initialize all the ranks to be the same one divided by how many pages there are in this case four.
So one fourth, one fourth one, four, two and four. There are 100 pages total.
Each will becomes 0.01. You're going to start somewhere. It's an iteration in machine learning.
Initialize all the weights to be what? In a big, classic deep learning neural network.
What? What are your initialize all the weights with? There's a problem with libraries like Partners Carousel that nobody looks at the code.
Okay. You're initialize them to Python random numbers between minus one and one complete BS where you took a complete crap.
But the idea is back propagation will gradually make it to be like real witchcraft.
So likewise initialize them here all to equal because in the absence of any link information right,
you will assume that every page is linked equally to all the other pages.

Hey, what is that called in here in my example here. Right in the for link example.

Okay. Suppose I have four pages where I link each page out, meaning I should have equal to pointing to the point that I link to the other three.

Right. But so do they. They link to me to link here.

So it becomes this. Everybody links to the other three. What is that graph called?

It's called. Yeah. A click.

Exactly. A click as a fully connected graphic.

We don't know that. So we assume that is a click. So we just simply make everything and we want the.

All right. Click Detection bar is one of the most complex meaning in terms of complexity, complex algorithms in the entire world.

You know, if Facebook wonders how many groups of WhatsApp users have 100 people in them,

how many groups have Facebook users have 10,000 people in that group?

Good luck with that search. You basically have to do a brute force search, believe it or not.

Okay. Yeah, because it's an example of a click.

Well, by the way, what I mean is in the hundreds, they're all pointing to each other like people in a class.

They are like following generic dialog. Bottom has no shortcuts at all.

All right, so then what do you do? So you update for a webpage.

Update trying to be the sum of each vs, rank the previous slide and divide it by a total number.

That never changes. So in other words that initially they're all the same number like one fourth, but divide that by the outgoing count.

It didn't work just for me. But what if for other people also how many total sum them for all the other pages?

That is not me. Okay.

So for each say there are ten in my example, I'll do nine summations because there are nine other pages potentially pointing to me.

But so what? All the other nine pages, each one of them will take the other nine and do the same calculation.

So we all have a first iteration of preview, but that's now a new starting point where the one divided by an equal is now replaced by not 1 to 11.

What made the change? That made the change? Okay, but that is still not the ending point where two then do it again, meaning take this and put it in here and then for every page, take that value updated is simply a value of ten.

So you watch ten numbers change after well, all the ten numbers stop changing.

That is a bridge. I'll show you a which coming up. Okay, so this is what I just explain, okay?

It's so easy, right? As an example here, you know, initially.

So suppose this example given to you initially,

all of the pages ranks equal one divided by five Y because there are five nodes after only one iteration,

some of them have become like much better than others. This like now 0.1.

This is 0.05. And this one is actually slightly less than slightly more than half, I should say slightly less than half.

You get three, right? So they're all not the same, although only one of them for you.

Okay. So how do we go from one divided by 5 to 1 over 20?

So a few one way or one over 20? Who's going to tell me?

Tell me why. One over 20. It's actually in here, right?

One link from poetry. One link from poetry.

So then poetry unfortunately didn't work just for this one, but also water for this and water for this and waited for this.

Wow. Said only give me one fourth of what, so to speak.

Right. So then you take the one divided by five initially that you had, but now do it before.

Why before again? Because three has four outgoing links and that is only what we're getting.

So we don't have to worry about what I was doing or if I was doing or for doing.

That is how one became 120. You can do the rest at home on your and you can do it now.

It's the same idea applied systematically to all the others. Okay then.

Now this will replace this. So throw this away now take this to new president,

but not the final prisoner swap supported one divided by 20 by saying one divided by 20 is now the new one.

But then what is Peter going to contribute? And then what? Like Peter has one over five.

So then what is Pete three? So what is three and one going to contribute to?

They can go on changing for a while, but after a while they will stop changing.

Okay, look here. After two iterations, you know, she is systematically going from it.
Initially, was this okay here they're doing the P, for example. So how did people go?
You know, the seven over 20 this the first iteration?
Interesting. Yeah. People. The Sun says after two iterations.
Right. But the calculation seems to show only the previous iteration.
Well, you know, you can update this for this. Okay. You can see how this changes to the place.
If you're doing it, do it if you want. So I'm not focusing on the steps, but I'm focusing on that.
So say you stop after two iterations in this, I'm done. Then what is the biggest number?
16 or 40 or the second 15 over and then cyber three over one over.
Literally that sorted. Ordering. Highest. Lower.
Lower. Lower. Lower. Rankings. One, two, three or four.
So what that means is if somebody searches that page, it's going to be a third first.
P5 is going to be sort of the top because it's more important. You can see why.
Okay. Of all of them. P5 has three links coming in.
One, two, three, one has only one link coming in, two has only two links coming in five also only two.
Four also has only two. All of them are either two or one.
But suddenly here we have three links coming in. The most popular person.
Higher strength. Okay, that's interesting. Twyla Easy.
Okay, so we don't have to go through all of them. Same idea. Okay.
So initially, I assume they're all one force, but this technically should have zero rank, right?
Because nobody's voting for it. Likewise, I should be zero for these also.
But we have a hack where, like a minimum basic income,
even if some sites have no links coming into them, we're still giving them a little start operation.
And yeah, president could never be zero. There's all kinds of tricks you can play with this.
Okay, so I'll show you one and fix. You have an artificial number called like all fighting.
Then put in like a number like 0.15 for no reason other than to not live with zero.
Yeah, like in this case. In other words,
what I'm saying is the page rank for this would be 100% rank for this should be 0002000 plus one in terms of fraction probably is still one.
Right. But that's not what will happen. These three will have the same number.
That is not one. This will have this one number that is not 1.0 be slightly lower.
In fact, it'll be lower by this, plus this, plus this. Imagine this, plus this, plus this subtracted from one.
That's actually what is going to help. All right. Select this.
Okay. Initially, 0.25 out right now. Ha!
Okay. So basically all have okay. Separation and is going to be 0.75.
Right. Because you are .25 plus .05 plus .24 and they all voted only for one.
So you divide .25 over one plus .25 over one plus point two, five or one.
So clearly, that is how we got to that point anyway. But what about BCD?
You're going to end up with zero, right? Please don't cry, you know.
Yeah. What's something do? Yeah. Consolation prize. We're going to do that.
Okay. So you want to generalize it.
So the generalized algorithm has an extra fudge value, which we call simply.
Actually. Wait. It's not here yet. Okay. So this is the classic algorithm, you know, right.
Where you simply do this. Like, what is not? But what about this number called the OC?
So this D is simply a number that you come up with like 0.85.
That way it is not 100%. Just simply page rank or total president got all but multiplied by point in fact, so that some other sites can get 0.15.
Wow. So fudge factor. Why is 0.85?
Trial and error. That's right. 0.82. That right?
0.810.89.85. Worked out best.
Okay. Purely just, you know, trial and error. And there's statistics that people right.
Where to modify the number and see what happens to rankings. Okay. Interesting.
I think I might show you one of those. It might be in the link.

But anyway, so the difference between the previous slides and this one is here we multiply the PageRank,
the summation actually by a value which is 0.85 for no relevant reason.
It's arbitrary. In fact, the reason might be other pages that have no links coming in to them might be one minus D divided by how many pages?
In our case, it'll be one minus three divided by three.
Like 0.1. Five divided by three is what each other's ABCs would get.
So all of these would get .15 over three, which is .05.05.05.5 total adding up 2.15.
And here it's 0.85. Yeah. Cool.
Okay. So there's that.
And then. Yeah, you can read this. Okay. It's exactly what I told you.
Okay. Yeah. That French factor so that somebody else will get their point on file.
Okay, great. And then have this whole iteration, right?
My rank depends on your rank, but initially nobody knows anybody's rank. So we start by one over and so many words for the same thing, you guys.
All right. Yeah. So then the proper use for the iteration would be eigenvectors.
Okay. But I'll show you a very simple JavaScript code that doesn't use eigenvectors, but you can use linear algebra and make it a little bit nicer.
But that the core idea is what I told you, that the translation where you update rank over and over until the ranks converge to settle.
So no need to update. You know when to stop. Okay. Like. Okay.
So number of iterations, you know. Yeah.
So initially. Right. Hmm. Well, yeah.
So this one is actually for how much difference between them. Okay, so suppose you have like, 322 million links.
Okay, then initially, this big difference between, like all the previous rankings,
but gradually converge, like all of the rankings were not changed from one iteration to under iteration.
They only differ so much. Likewise, if you start with the fewer pages, slightly bigger difference.
But the idea is through many iterations, successive iterations will converge.
Okay. The gap between them is getting smaller and smaller. Okay.
Yeah. Okay. What about this? This way. Interesting, right? You would think that they'll end up with the page rank of even a half, Correct?
Because of the click that I showed you is a four node click button or click will be A pointing to b, B pointing back to A, Yeah.
So then that can actually be the initial value should be 0.40.5.
Cool. But what about with the 0.8 file is a cool thing you can do at home.
If you want the grade here, it'll be half half without the fighting factor of D,
but if you have a French factor you can wonder what should be the starting point, What should be generation gap with it?
Okay, See that? That's very interesting that suddenly it's not just very simply half, half converging right away.
It's because of the point that favorite coverage after a while.
But after a while they will converge. By the way, it's actually very cool.
And yeah, this little spreadsheet that is showing you after 20 iterations, right?
The book got a page rank of one.
By the way, page ranks can be normalized, are not normalized, can not normalize projects can be anything even like one each.
But if you want to normalize it, then obviously would make half each okay. But the idea is to both converge to the same value.
It's roughly the similar y slight difference because of the 0.85.
And maybe if you do more iterations, it'll actually converge. Okay. But do you see how different there are initially?
In other words, what I want you to be surprised by is that big difference.
Even though the page has high symmetry since the pages are extremely symmetry, one should not be higher than the other at all.
But because of the convergence algorithm we use step by step and the 0.85.
Initially they are different, but gradually they'll become the same. And in this case.

Okay. So that's what this conversation was about, right?
Okay. And then one more example. Right. You know.
Well, yeah. So I guess three to this become one point again, same thing.
The number can be more than one. You can normalize them afterwards, but the numbers are converging.
Okay. They're confusing. Right?
I want you to maybe read this and go on Piazza and then post me your confusion, if you like, or read this other words when.
Basically, Alice made these slides. He was a little bit sloppy with terminology.
He's a great guy, by the way. I have nothing bad to say about him. But then, you know, the calculation is a little bit loose.
So when you do this carefully with the calculator, we'll see. Oh, my God.
Why? Because the honest answer is your initial starting point, right?
It doesn't matter what you start with. The best starting point is one divided by ten.
Right. But say you randomly pick starting points, actually, literally, randomly, even then, they will all converge to the same value.
That's very surprising. Okay. Complete random numbers. You can try it. How would you try it?
I'm going to give you a code. The code is coming up. In fact, here it is.
Oh, okay. You see? Okay, So first I'm showing you the result where one one.
I start with one. One and the end of it. One, one. Okay, four. But no, it's pointing to each other.
What if you start randomly? In fact, are actually this one?
What if you start with point one? 4.4 are What if you and make them entirely different?
Look at this. You know, point one, 5.2 cents, and you can basically set anything with anything.
Right. See this? They start them all with one one.
I start them all with zero zero, but I start them both with two different values.
Point on four points and then you get the same.
And so which is what the models in the actual numerical value doesn't matter because we normalize everybody.
In other words, this will become half. Half. This will also become half half.
This will also become half of the algorithm half hour after you normalize it.
So where is this magic code? The code would be semi equal to.
PR. We're going to try it. Look.
Okay, take that an inch and change x equal to page rank PR hit enter.
Well. All right. So we have four nodes and the nodes are represented.
They're actually six nodes. Okay. Imagine there are six nodes. One, two, three, four, five, six.
Imagine the index. 035. One, two, three, four, four.
All that is simply specifies is what is connected to what node number zero is connected to 123 naught zero is connected to 123 because initially zero,
right. This is 01234, five.
That's where the nodes are numbered. A second node, which is node number one, is connected to one four,
not number one that's connected to two and two for that idea because you can do the rest.
So it is not an even distribution obviously. So then we wonder what will happen to the president.
Obviously, you start with same value for all the presidents, but see how they end up.
Okay. Then the conversion values would be those.
They're all no different. And the difference is because of how they are linked together.
So you can play with them. In fact, what I did actually was this one.
So you can delete all of this. I mean, all of these just have one nodes that all you need delete.
Now suddenly you have six of them. So each is 163.166 quote, and then they will converge and become something different.
And why are they all different anyway? Because this graph is not a symmetry group.
You can do anything you want. Suppose this stops working for one.
It only works for zero and three and that'll change the page rank.
You start with the same one six to all always. But in the end you will get something different.
Okay, it is neat so you can try and make one more change.
The last node, which is not number five, likes the very first. Not only what for zero and that will change the baseline.
In fact, that made zero command to be the leader. Wow.

Interesting. Yeah, actually, no.
The last one became the leader. I don't know how to go on, like, sure what is happening again.
But the idea is I can change anything as long as the values are legal to say, say number zero is going to work for all the others.
So it's going to look like what leaders. So anytime you make a change, it will change these numbers.
So I want you to play with it. Where's the code for it? It's right here.
Here. It's not very big, by the way. Okay.
And that iteration function just on that is making the bridge event changes.
All right. Anything else I want to tell you for a moment?
It's okay. President conversions.
Yeah. So the whole damping factor that is called.
Okay, so this is simply a trial and error number that is basically set so that you don't have to iterate like a lot.
If you said database to smaller to large, let's say the 0.95 equals 0.5,
you will find that you have to iterate many more times to get the convergence,
whereas with 0.85 is least number of iterations to come to some kind of convergence.
Okay. So the fact that by trial and error like that, well, okay, it's all a classic solving like a recurrence relation.
Okay. In matter. All right. Or in machine learning, that's like a learning rate.
Okay. If you set the learning rate to be too low, it's massive number of calculations.
Okay. That data is like very slow set learning rate. Too high. What happens?
What happens in a neural network back propagation when they set the learning rate to be extremely high because they are too impatient or they know,
step through my covers and what will happen? Convergence will become divergent sexual and exploratory scale exploding.
Then, you know, the rates would never converge. Same thing can happen here.
Isolate. Okay. All right, then this all just more examples of over and over again.
You've got these sprays, writings and observations. Yeah. It is also very interesting because there are no links at all completely throw it out.
They're not going to be useful for anything at all. We don't really know what actually Google does unless you are critical.
These are all guesses. Okay. Okay.
So likewise, in many real sites, this actually the case right where the home page might have an about page and have a product page,
product page by might link to a whole bunch of products and then links to external
sites might link to your competition in order to write your Yelp reviews. So you can have a hierarchy regardless of what the pages are.
The president algorithm keeps working over and over. That's all. Yeah. So we don't have to worry about example one, two, three.
They all do the same thing. There's really nothing to tell you, actually.
Honestly. Okay. Yeah. And then again, say, given in a situation like this home,
because so many things linked to home, home ends up being like a high page ranking and so on.
And the average always works out to one like it should be. All right.
Yeah. One more hierarchy, a very simple hierarchy. Just, you know, you do this and react, right?
Make a flask in the back and you make it. That's what all this is.
So when you set up links between pages, how will that translate the page amount?
You can see when you make a home page and then it's link into three other pages.
They're all only linked back to you. They'll all have equal links. You will have a higher link.
Why? Because they're all being wanted only by you. Each one is getting one vote.
You, on the other hand, are getting three votes. It's the same intuition over and over again.
Okay. Two more minutes. Okay. I want to finish a little bit more.
I'm determined. Okay, so then same thing, right? You have something called site A, some external site that is pointing to a home page.
And then home was doing like all of this, you know. Okay. Here are just some numbers, really.
Looping. Looping is very interesting, right? So what happens when Page is technically linking this word link forms?
Divider can in a link form, people artificially make a site?

Okay, well, they might have 100 pages and they all link to each other like in a loop.
Okay. Then what should happen? You know, you still get like all the ranks to be equal.
It all comes out to be even anyway.
So you can cut, you can make a graph, you can go to my show that I showed you and make this be the graph for everybody.
What for the next person. Each one watch Exactly. Only once. See what the president can register.
That converges to equal. Okay. Please try all this. Really? There's a little hint for you.
You must try. Okay. And then they are just different structures.
When you make a different structure, like with a loop, but one coming in, one going out, Then what happens to my even numbers is not even anymore.
She's not even. So the point of all of this is the structure of the page.
The structure of the graph obviously has an effect on the ranking.
It's all asymmetric graph, symmetric rings, asymmetric symmetry broken.
Okay, that's it. Again. Why this one? Why did this increase?
Because this is one more link coming in. Of all of them, there's still only one or two links coming in to see all the others only have one link.
And why is this the lowest? Because it is actually being pointed to only by this one site.
So same with this one. All right, enough of all this.
Oh, yeah. Then again, more. Now you play the whole click game with this, okay?
Obviously, you're going to get exactly to be equal to one. Okay.
And then you have the same as before. But now suddenly we're not winning.
This is getting a little boring, right? You understand? On some intuitive level, you understand all this.
Okay. It's too easy. Hang on. Don't go.
Please don't go. Same here. The one that is linked to the most is going to get the highest rank you can go see.
Okay, look, why these little differences? And then you look at this again.
Thousand incoming links, only one outgoing link and also a spam page might do weird things like that, right?
It's not going to be rewarded. So spam got punished. Zero cases passed from God punished.
So good page will always win. Okay, so a real reputable page will always have appropriate spam pages these days.
Okay. This is what Rich was asking. You see this? So you simply make a bunch of pages.
All of you link to CNN dot com. You know, then somebody would think you're important.
Google will think are important, but Google does not think you're important. Still only 20.
Yeah. So what you can do really is some structural improvements you can do,
but really the very best way to increase your page rank is this provide high quality content.
And why does it matter? Because other people are linked to you. Yeah, that's basically what Google tells people.
Okay, so this premise looks every new iteration is lose.
So if you want higher PageRank, provide good content that people want, but we'll finish with this.
There's only one more slide. See this these days, right?
It is not true. It's all about the links. Okay.
It is actually about the content to go through your page and see what kind of content you have, you know, and what kind of site are you.
So all of that well, you please read this amazing site, this page called Praise Language video.
It tells you what Google actually might be doing.
So that last one, you can then what it is, is this okay, so that simple algorithm and launch this massive empire called Google.
But now they pulled back and said, that's not all there is. We can do more code things.
Okay. All right, guys, Sorry to keep you a little bit longer, but we got through even more.
So I'm behind by one topic. Okay? Just only one topic. We will catch up.
You're welcome.

Lecture - 2
Okay? We'll save it for a break. Okay. Um, so one quick announcement.
0:01
I have a call to make to China at 9:00, so we'll not have, like, two long office hours.
0:05
Okay? Have to rush back to my office, I apologize. I wanted to tell you in advance.
0:11
Look at how many people are missing. I'll take attendance over and over.
0:15
And when we find people that are not here. Oh, my God, they lose five points.
0:19
It means more for you. Wow. Glad you came to zero sum game.
0:23
Okay. Oh, uh, it's all relative at the end of the term.
0:27
And so if somebody loses five points. Oh, no. Oh, yeah.
0:31
You should have text them and have them come to class. Hey check it out.
0:36
So I'm going to start at the top. Also we.
0:39
Will not have a. You know, too many discussions about question four.
0:45
Question five, all that right for the midterm.
0:51
So we talked and we made two rubrics be a little bit lenient obviously because there were more than one interpretation.
0:53
Right. There was more than one interpretation. But it cannot relax it so much that almost anything goes.
0:59
Basically no, that's not the point. So please understand the fine distinction I'm trying to make.
1:05
In other words, we cannot basically say yes to every request where no matter what you wrote.
1:11
Wow. And I want to get points. So try and be a little bit more precise okay.
1:16
For a final it'll be hopefully easier. I didn't think the midterm was hard okay.
1:21
I never mean that. But maybe there were too many questions. Okay, so I'll try and cut down the number of questions.
1:25
Um, try to make it more objective. As much as I can not be too open ended, if it is open ended, I will tell you it's very open ended.
1:30
You know, you can write, like, a lot of different things. Okay, so we can definitely make it.
1:38
I want to help you basically. Okay. Not make people, like, get bad grades or anything.
1:42
Last time in this class, many, many, many students got a history of minuses.
1:46
So you shouldn't worry about the grades. Just focus on hopefully the fun things you're learning.
1:50
These things are stuff that everybody has used in the world, including you and me.
1:55
You know, even ten years ago that was Google, right? But this course basically fills in the blank, fill in the blank,

1:59
fills in the blank for you so that when you're doing research you'll know, like how all the images are being served to you today.
2:04
Autocomplete. So I'll do some autocomplete. How does it know what to fill in.
2:10
So these are useful things to know because in the future you can use them with large language models.
2:14
And how I do even more of like what we show here. So none of this is obsoleted.
2:19
It just you can build, you know, for the future. Even more so.
2:23
On that note, we have two more homeworks, which I'll give you very soon by the end of this week, most likely.
2:27
Okay, because I want you to have about two weeks for each homework.
2:33
You don't need two weeks.
2:36
The very short homeworks compared to the first three, the last two are much shorter now, almost like a tutorial you do step by step by step.
2:37
It will work okay, but I want you to experience the fun and magic of things like retrieval argumentation.
2:44
I told you over and over, right? A few things survive in this whole big churn of things that are happening 2 or 3 days ago.
2:50
Stability AI, the name has the word stability in a stable, very stable.
2:58
But it became a pretty unstable company because its founder and CEO, Mushtaq, was forced out by the board.
3:03
He actually resigned. Okay, she has nowhere to go. He might do something else.
3:10
But there is a company that makes the image generation in our eye possible.
3:13
But, you know, something happened, obviously. So I started to live. So we never know how this will all play out in the next few years.
3:17
Okay. And there's lots of lawsuits.
3:24
So many content creators here in New York Times, this show, OpenAI saying you basically crawled in our site and took every news article that we had.
3:26
And now ChatGPT before is able to answer what New York Times subscribers pay money for.
3:33
But then you get the money, the pro account gets the money. So OpenAI needs to pay many millions of dollars fine.
3:38
Still, in the New York Times, that is not settled yet, by the way. Okay, so there's all kinds of things that happen behind the back, right?
3:44
But you have the open source movement from the open source movement.
3:50
The large language model source code is available to us, and there are almost 300,000.
3:54
These are not small numbers. I talk about almost 300,000 so called pre-trained models.

3:59
Pre-trained models means they already have the training taking place. They are ready to basically not do tasks 300,000.
4:05
So there's lots of efforts by companies, not open source companies, to build products that will use the open source LMS and not OpenAI,
4:11
you know, GPT five, okay, not Gemini, because Google will never let you near the Gemini in know code, right?
4:20
Or that the train that you can only look at it from the outside like a black box.
4:27
But if you have source code, tell lamps tens of thousands of them, hundreds of thousands of them, then you can do more things with them.
4:31
So that is the kind of homework I want to make for you. Okay.
4:37
Also, one more crazy cool thing is happening, which is the alarm is almost becoming like a small function call.
4:40
So when you write a piece of software,
4:46
you don't have one main function in which you write 1 million lines of code and just run the main function, right?
4:48
That is what prompt engineering look like so far where somebody wrote this massive bunch of problems.
4:53
But it's all in one big prompt. And then it came from programing languages where people said, we cannot write large promise by hand.
4:58
Let's actually inject things into our prompt programmatically. But it's still one long problem.
5:05
But now that is changing people actually talking about mixtures of experts, Asian programing like how many different agents,
5:09
many different and experts and then they all collectively solve some problem together.
5:17
It means agents can pass data from one Asian to another Asian.
5:21
But who's doing the programing? We are actually doing the programing. So that's actually very interesting.
5:25
You know. So these are all Lagrangian very crazy directions because also one more which is actually is pretty surprising.
5:29
It actually says how to put how to merge. It's called model merging.
5:35
How to model merge 100,000 models together.
5:39
What the [INAUDIBLE]. What does that even mean. Because are hugging face.
5:43
Any proof has about 300,000 models. Here somebody wants to use each model for something and build this amazing mega model.
5:47
But still, it's piece by piece. Okay, it's never combined in any real way,
5:55
so I like that a lot because I like traditional programing so suddenly are not making a Java function call and making a lambda function call.

5:59
And then Nvidia comes along and says,
6:07
we can take all this lambda function calls and run them on black wall GPUs on the cloud somewhere and make them into microservices.
6:08
So then we simply call a bunch of those microservices, like, do you know, make any traditional application.
6:15
So these are all like taking off in like strange, crazy, cool directions.
6:20
But we become more and more empowered in the meanwhile, though, you know, traditional knowledge that you have hard one about C plus plus Java.
6:24
They will slowly become like obsolete and rusty. C plus plus is not 100% going to go away.
6:31
COBOL never went away. But what we mean is large automation people won't be needed right out anymore.
6:36
Why not? I can do it anyway. So. So there are two homeworks that I have for you would be about those.
6:42
Okay. It will be something so cutting edge it will be totally ready.
6:48
In fact, believe it or not I am going to try and use them. So Namaste.
6:51
What was announced last Tuesday. Now to make like one of those two homeworks.
6:55
So then you will have 100% container of microservices named GPU,
6:59
because you get 60 days of GPU hours per frame from Nvidia in officially to do the homework.
7:03
And I'll get a taste of it up here. You're going to become hooked.
7:09
Now, once you see how powerful and easy it is to write applications of Ohmygod, that is exactly what Nvidia wants.
7:12
So we're not there yet. But give me some time. Okay, I haven't made it yet.
7:18
Uh, what else should I tell you? Today we have three topics and we can finish them.
7:23
They're all very small. I put them right here. Roughly an hour each.
7:27
You know, I always say that, but you know it another time because I have so much to tell you.
7:30
But these are all hopefully relatively simple. A few line description is what this looks like.
7:34
Snippets are just simply highlights.
7:40
Highlights. Meaning you do an image search or type right now Dreamworks animation,
7:43
then suddenly it is not giving you just the URLs and PDF files that mention the word Dreamworks animation.
7:47
It actually gives you a picture of the Dreamworks campus, and it says Dreamworks animation.
7:53
Founded in 1995. Founder Steven Spielberg. Katzenberg in a different word, one says, making a summary for you.
7:57

Write that summaries promptly at the very top, and if you like it, you don't have to look anymore.
8:02
That is called a snippet. Snipping means a piece.
8:07
So and whereas the piece come from the piece can actually come from one of the pages they're serving you.
8:10
In fact, the piece that snippet highlight can come from a page that you wrote.
8:15
They even give you advice on if you want your page to be pulled out for a snippet.
8:19
How should you structure your page so they are not making the snippets? Okay, actually, coming from what's already in the pages.
8:23
The next step obviously, is Gemini OpenAI. The chat itself will summarize what's in the pages.
8:29
That's a different kind of snippet. Okay, that's a search summary. So these snippets are coming from existing URLs.
8:35
But then they pull it out and put it at the top. So you don't have to click on the URL.
8:40
That's useful okay. And then they have rich media snippets. That means if I type like a hub or something it'll show me about video links.
8:45
I can immediately click on see image links, video link, Spotify playlist links, Taylor Swift Mary song playlist.
8:51
We call them rich snippet. The word rich always has meant multimedia.
8:58
So rich means anything that is not text, which is only three things audio, video,
9:02
still images like photographs, artwork, you know, 23 okay, so we call them snippets.
9:07
Okay. Um, then query processing.
9:12
So you already know what query processing. Right. So what happens when you do a search.
9:15
It goes in the search index. And then it looks at the URLs and then does page rank and gives it to you.
9:19
This lecture is a little bit more um fine detail, but they cannot possibly look through every single URL at every index.
9:25
So what are they allowed to throw away or commonly search terms?
9:32
Maybe like Donald Trump, you know, um, they don't have to go to the inverted index.
9:36
Maybe they can catch them. That means they can store them in high speed machines.
9:40
And as soon as somebody else searches for the same thing, Baltimore Bridge crash,
9:43
so many people are going to type Baltimore Bridge crash, so they don't have to keep going back to the index.
9:48
They can store the results and quickly give it to you because it's the same search, right?
9:52
So they can monitor what people are searching for. Twitter does that to write trending tweets okay.
9:56
Are trending YouTube videos trending TikTok videos. So they have trending merit trending Google searches.

10:00
So that is the whole query processing stuff. Then we also have autocorrect autocomplete two slightly different but related things.
10:06
Autocorrect is when you make a typo spelling mistake, it fixes it.
10:15
That's a very easy, fun, dynamic program algorithm called the Levenshtein algorithm.
10:18
And I have a little JavaScript implementation I can show you. You can play with that, but I'll just verbally walk you through it or um,
10:23
go through the triple for loop line by line, you know, or a um, that is autocorrect.
10:28
Autocomplete is when you type in a few words, most likely it can guess what the rest of the words are going to be,
10:34
because so many others of typed the same words and they all wanted the same completion that it's going to suggest to you.
10:42
Um, again, it's a form of recommendation that way, which saves your time.
10:47
It's a drop down. And then if your completion is in the drop down, pick it, then it saves your typing.
10:51
But if you're typing, if you are typing something very new, it's not in the dropdown.
10:56
That's okay. Ignore the recommendations and type it in. And if enough people do that, it will then be added to the recommendations.
10:59
So they're dynamically doing lots of work, you know, processing in. Otherwise you're looking at your own queries throughout the world,
11:05
global queries and helping us next time to make that same query be, uh, served faster.
11:10
All right. So these are all like pretty easy, right? There's nothing magical in any of them.
11:17
The autocorrect like I said, Loewenstein is a nice algorithm.
11:21
Loewenstein algorithm came from bioinformatics, which actually used to basically fix gene sequencing errors.
11:23
Okay. So incredibly it found its use in NLP.
11:30
It found it used in autocorrect. Yeah. So autocorrect came from bioinformatics I can even tell you about that.
11:34
All right. So we uh talked about this talked about this I only have one new thing I operating system.
11:41
So some people actually think that it's time to make an LM Foundation operating system.
11:49
So you have the actual traditional operating system like Mac, you know, like windows.
11:54
Linux would run. Right? And you have the hardware here, actual bare metal hardware.
11:58
What about on top of that, an LM operating system.
12:02
So LM calls. So the operating system is going to expose some calls just like standard windows OS exposes calls to

application developers.
12:06
Right. Chrome zoom, all that. Likewise, what if the L am operating system layer exposes some calls to, uh.
12:15
Apps, which are things like ChatGPT or a recommendation system, document summarization, document translation, all the things you want to do, okay.
12:23
Or even, uh, um, entity recognition.
12:31
So then you make calls to those, uh, kernel API calls and then the l'Allemagne turn obviously is going to make calls to the underlying voice.
12:34
So that is the idea I'm going to show you the call it AI os, but it's actually a lemos okay.
12:43
Look at this. So that's the operating system. And this is the hardware layer at the bottom.
12:47
And so then you have and that's the actual OS right. The words that I have here.
12:52
But look at the OS is actually interfacing this OS interfacing with the LM kernel.
12:56
So this is a new thing that this paper is proposing. And the paper actually the kernel writer has all this functionality built into it.
13:01
The traditional operating system has lots of functionality built into it. Right.
13:09
You can, uh, you know, run multiple processes. You can start and stop.
13:12
You can ask for disk space, you know, you know, like you can do input output, you can do port connections.
13:16
So all those things are operating system functions like West L am operating system functions or about agents and memory and storage and all.
13:20
And then the top calls are the ones there's an API here.
13:28
And then you would make real world applications. Now for example a coding agent.
13:32
So this is the vision that these people have. They wrote this entirely in Python.
13:36
These three people check it out. It's actually very cool. Okay.
13:40
Everything that I am telling you is brand new. This stuff yesterday. Okay. Got last year.
13:46
So yesterday l l am agent operating system.
13:49
And again agent just simply means one function call. You know, 1LL am this one thing.
13:53
So the idea is we'll write code so that agents are, like hooked together, like a function calls all hooked together in an app.
13:58
So same idea. It's very cool. So you can read this afterwards.
14:04
But there it is. Right. See here travel agent.
14:08

For example, a travel agent would need to, you know, do all this write flight recommendation idea, upload seat selection.
14:10
That can all be done by different parts of an LM or even different LMS actually.
14:16
And finally, the travel agent L am a travel agent.
14:21
I is actually able to book this San Francisco generic trip flight for you, while for me you could even 20 years ago,
14:25
you would have to go to a brick and mortar place so called travel agent,
14:34
and they had access to a database called a saber database, which is all the airlines reservation system.
14:38
You and I had no access to saber. Sabry saber, okay.
14:43
Only they had expensive computers and got there. So you pay them money like some travel agencies so you can buy your ticket to come to the US.
14:47
But now, as you know, you can do it yourself. You can go to Travelocity and Expedia.
14:53
All right. But now this the next step. Why should you even go to Travelocity?
14:57
Waste time. Just give it an hour. What do you want the A anyway?
15:00
Okay, so it's interesting how there is that layer that is being proposed.
15:04
And here it is to talk about OS history now okay. That's a very neat things in here.
15:09
And so I think you should read.
15:14
But the arts of Asian programing, you see so interesting again you know, determine whether there's going to be a reign.
15:16
You know that's so cool. So you booked a flight, but then you also want to know if it's going to rain.
15:23
And so then you asked the weather API at some point you see at some point you go back and then you go to the internet and ask weather.com,
15:28
but do you manually do not do it yourself, but in the meanwhile is involved this idea okay.
15:34
Great. So then if Nvidia has their way, all of this would be named function calls plus you.
15:41
So this will all the microservice regional piece would be a microservice wrapped up in a name layer like a Docker container.
15:47
But the best part is the name calls are already deployed in multi-GPU clouds.
15:53
You don't even know where they are. Your app doesn't know where it is. It just makes a call.
15:58
Write a function call and then it runs somewhere in GPUs super fast.
16:01
So it's the best of all worlds for all of us. Call.
16:05
And it's all like Python. Do you see that? Okay. So that is the actual large language model.
16:09

So Gamma or GEMA uh from Google.
16:14
So Google also contributed a small language. I call them small language models to be stands for 2 billion parameters.
16:18
2 billion. Sounds like a lot right? It's actually not chargeable 4.5 has 1 trillion parameters which is 1000 billion parameters.
16:25
So this only has 2 billion. Actually 2 billion is the smallest that I've ever seen.
16:33
There's many. There are 7 billion. 14 billion okay. 2 billion is tiny.
16:36
And 2 billion might be five gigabytes. You can download them on your flash drive and you have your whole.
16:40
So where do you find them? You find them in Huggingface. So when I go there, I'm going to say Huggingface models model repository.
16:45
You should be in the habit of going here. Wow.
16:55
Half million models. Okay. And each model is already pre-trained.
16:59
See, there's also Gemma and Gemma again. But now it's a 7 billion parameter model okay.
17:03
And then Facebook contributor llama. Okay. We are not time to go to all of them that came from France Inria, a French research consortium.
17:07
There are so many now. Cohere is also an AI company in San Francisco.
17:15
So they made that one three days ago. These things are like extremely new, right?
17:19
Two days ago. Seven days ago. But any one of them can do something useful for you.
17:22
Meaning all of those things on the left. They're all like AI tasks.
17:27
So the vision is our app can mix and match from the left hand side.
17:32
In other words, say I want to do text classification. I can click on here.
17:36
And any of these models can do text classification. They're all open source.
17:40
They're all free. I can just name one. I can even try five of them to see what works best for my application.
17:43
Okay. It's just incredible. And how do you download all of them?
17:48
How do you start using them? You can click on any one of them. And then there's course called a model card to give you a little summary.
17:52
And then they even tell you like here they give you an example somewhere you can go look at it okay.
17:58
I mean they actually tell you text classification. I'm going to say, uh, I love rainy weather.
18:03
Let's see what happens. Okay. I love rainy weather. In LA.
18:09
Okay. So you can try it out, but also give you some lines of Python code to import this and then actually try it yourself.

18:13
Okay. Um. See, I don't know what I'm looking at, honestly.
18:23
Or these are all like different languages. Even weirder. Okay, I guess this one is trying to actually infer what language.
18:30
Oh yeah. Okay, so I typed this in English. Right? So 99.2% sure that is English is doing language classification.
18:37
So you can try typing something in Chinese or Tamil to see if it recognized, okay.
18:44
It's not about the weather okay. Sure. Uh.
18:48
Was a maple city. Let's see what happens.
18:52
Oh French 99.5%. It's pretty cool, right?
19:04
Because that's actually a very big deal. Okay. You walk around at you, you hear some strange language you never heard before.
19:08
Your brain cannot automatically tell what language it is. Okay? It's not magic, right?
19:13
But then to the Dalai Lama, it sounds like magic because they have trained it on so many different things.
19:17
Suddenly it seems to know, like what? These words are very impressive anyway.
19:21
So they tell you like how to use it. Okay. And then you can then start doing it.
19:25
But the best way, in my opinion, is to actually go to this site called LM studio.
19:28
I mentioned this in class. Get them for you, get it for your operating system Mac, windows, Linux.
19:33
And that will then go on hugging face that I showed you and download the model and actually put it in your directory somewhere.
19:39
There's like an IDE for writing agent apps which you can use this ID, and this ID can one day hook up to Nvidia's, uh, microservices.
19:45
And they have this nice container and all of this that you write with these runs locally,
19:55
because you download the model to your computer and all the regulatory l'augmentation running locally.
19:59
So that way you can take it to employee ID or interview and actually, you know, show it to them.
20:04
You don't need internet connection. Yeah. So good question.
20:07
You know, you manually do it, you know. Yeah. Huggingface basically says, here's a gig model ticket.
20:18
I mean, you can click on any model that this UI makes it automatic. That's all.
20:22
You name it, just simply name the the model, it'll go and search for it.
20:26
They usually involve an extension called json l, json l.
20:29
And so then yeah, it just automates it, that's all. Otherwise yes you are right hugging face.

20:33
You have to manually get it yourself okay. It's usually a download somewhere.
20:37
You guys files and versions. I'll do just one of them for fun okay. See this 1.1 gig?
20:40
Actually, that's not the one. This one, uh, dot model, you know, file national.
20:45
Not the one. I'm sorry. This one. Hmm. These are all weird formats.
20:50
You know, I'm looking for, like, Json, l format. I'll tell you what I look for Lambda.
20:54
There's so many llama models. So many. We okay?
21:00
Okay, let's do that one. Transformers alumni, Facebook, uh, train deploy files and versions.
21:06
Okay. Oh, wow. See llama is not given to you for free.
21:13
They basically bug you a little bit. Yeah. Okay. So I'm not going to do that. But at some point you will get like about a ten gig file.
21:18
And the Json format is a text format. It means if you do a cat command Unix cat command on that ten gig file,
21:24
you will see all the English sentences on which the transformer was trained on.
21:31
It will actually see the raw data. That's quite incredible.
21:34
In OpenAI, there's no way in [INAUDIBLE] deliver and tell you what they used to to train on.
21:37
That's one of the problems actually, because they call themselves open, but it's not open okay.
21:41
Okay. So something. Something small. Huh? I'll try.
21:45
I'll try. Just one more. Mistral must probably be, you know, the.
21:49
Probably. Don't ask you for your credentials. Okay? Okay. Model card for Mistral and be, uh.
21:53
We're going to do it. Text generation, files generation. Show me a file.
21:58
That's all I want. There it is. Okay, these are all checkpoint files.
22:01
About 12 gig. Why don't we get one? Okay. Mhm.
22:05
Okay. Yeah. We should just do one on wireless downloading. I'll just let it download and just let you see.
22:10
For 12 gig you're able to get like an entire large language model.
22:15
Yeah. There's something revolutionary going on okay.
22:19
Obviously the closed source companies you know Microsoft, Google OpenAI does not want all this to be out in the open.
22:22
But you know, so potentially we're going to do okay, one last thing.
22:28
And then we're going to get the actual topic if rag is cool but I guess cool.

22:31
What about raft? Oh, a new acronym.
22:36
Okay. On the one hand, you have the large language model.
22:40
Which is trained on something broad and general. If you have access to the transformer source code, meaning you can run more training,
22:49
you can actually do extra training on some specific domain like a medical domain, legal domain.
22:56
It is called fine tuning. Then you can fine tune the model for your application.
23:02
Okay. That's one way to do it, to make it give you intelligent answers.
23:05
Or are. You can actually not if you are not able to do this because you don't have access to the code.
23:09
You can always use some external memory. We call it external memory because that could be a relational database, a Json file, csv file.
23:16
PDF file Mongo. Uh, you know, Json document, anything at all.
23:24
A vector databases. Then you can tell the LM through your chat query.
23:28
Don't answer the question. Go here and then get the answer from there.
23:33
And come back and summarize it to me in text form. Therefore, you basically bypass that alarm and say that is where the good answer lies.
23:37
That is called retrieval. Augmentation because you are retrieving it, but you are augmenting it by this extra round trip that you make.
23:44
The consider two different things, right? Fine tuning, ritual augmentation raft mix system raft is called retrieval.
23:51
Augmentation via fine tuning, because it turns out that it's still not this fine tuning.
23:57
You can actually fine tune this part so it and it answers even better, because sometimes rag actually fetches your bad answers,
24:03
which is very embarrassing because on the one hand, we know that rag itself contains good answers.
24:10
Okay. It definitely contains a high quality in answer that you want, but somehow the retrieval didn't fetch it.
24:15
So you don't care. You're still getting a hallucination. Basically bad answers, right? So raft is a way to fix it.
24:21
So raft will make rag urine more high quality gas from Microsoft, like about 3 or 4 days ago, I think, you know.
24:26
So let us look at raft. Okay. So Microsoft laughed.
24:32
You should all become familiar with this.
24:38
All of this that is happening. See this ritual? Augmented fine tuning.
24:42
Okay. March 15th. I lied 11 days ago. Huh?

24:46
Don't kill me, okay? Okay. Sorry. The cool. Right? So rough method.
24:50
Close. Examine the colors. Right. Yep. So I guess open source.
24:54
You can read this afterwards, but the whole idea is that it's going to make drag even more specific, meaning you want more high quality.
24:58
So it's amazing how, you know, like I said, all these different people, all these different groups,
25:06
they're all trying to, you know, make things even better than how you used to be.
25:11
All right. So you can go read about draft. So all that brings us to.
25:15
Now. I think I've gone through all of this. Yeah. So look at the autocorrect one, right?
25:22
Oh, I hate audio. Correct. And then this one, uh, autocorrect pass to a restaurant and.
25:26
Peace. Rest in peace. Right. And then my worst enemy.
25:31
My worst enemy. Okay, there's a whole bunch of autocorrect memes. Okay.
25:34
And this one, uh, I don't give an f. I don't give a duck.
25:38
Okay. I'm tired of your [INAUDIBLE]. Autocorrect.
25:42
All right. Oh. This one? Yes. You know, so this is also, like, super cool.
25:47
Actually almost forgotten. This is neat. Look at it. Okay.
25:51
So you know, when you search for images in Google, right? The Google image search, we do it in one of two ways.
25:55
Most of us type something like, you know, uh, a Welsh Corgi puppy text or brain puppy, I should do it.
26:00
Or which is actually what this is called text image alignment.
26:08
Okay. I will type Welsh Corgi puppies. I used to have one.
26:11
His name was Pippin. Oh, cute. They're the world's smartest dogs.
26:15
Oh my God. What do like 20 years okay.
26:19
Okay. So then you see, it obviously worked, right? They're all like Volkswagen puppies, right?
26:23
Almost all of them, I guess. Okay, so that is one way that the search engine works from text.
26:27
It is bringing you relevant images because the search engine was trained on words like Welsh Corgi and those pictures together,
26:33
that is called combined training,
26:41
like multimodal training in a way so that the words will then pick the appropriate pictures or actually to end and trained in a very different way,

26:42
which is trained on image pairs. Meaning if I output my own corgi and say show me more dogs like this, that is called image based search, right?
26:51
A Google Lens, you know what going to call it. So that is also going to produce very similar images.
26:58
So that is an image to image in our training okay. So this picture that I am showing you uh handles both cases.
27:02
See in here. That it says this one is image.
27:08
Image alignment okay. So that means to extract what are common to dogs okay.
27:12
So dogs are very similar characteristic nose and ankle. And so then all the dogs would then go to a certain point in the embedding space.
27:17
And the similarities between them would be like very high similarity between a dog and uh, duck would be extremely low.
27:25
So punish the model if it produces similarity. The point of all of this is this.
27:33
Okay, you'll need to train a layer called an embedding layer, which is a neural network.
27:37
So the idea is to have this embedding neural network. And then you have this abstract embedding space okay.
27:42
So embedding just simply means a bunch of numbers. If you want to.
27:46
If you want all the dogs to, you know, be together. For example. Okay, you pass a dog picture and then have it embed somewhere.
27:50
You also have previously labeled training data like you know where dog should go, so to speak.
27:56
Okay, so imagine you made high quality dog embeddings, right?
28:00
So then you know that the dog embedding should all go over here to give it a new dog, a picture and say make an embedding for it.
28:04
If the neural network makes an embedding, which is a list of numbers, and sends the embedding to here, that's a very large gap between them, right?
28:09
Similarity. So then you punish it,
28:16
meaning you backpropagate the error so that next time it'll produce the embedding to be slightly closer backpropagate more time,
28:18
it'll then make all the dogs go here. Likewise, the same neural network can make all the cats go here like a multi-class labeling.
28:25
Okay, but it's not labeling. It's producing embeddings as output that is called the embedding layer or the embedding model.
28:32
So that is what all this about okay.
28:38
So Google has obviously trained, you know, the giant neural networks on embedding models for almost anything water

bottles, cups and all that.
28:39
Right? In two different ways using images themselves and using text, the actual text.
28:46
When you say water bottle, you know, make sure that you produce like a water bottle. All right.
28:51
So then all that is explained very beautifully here. You can go and look at it.
28:55
So then after all that is done the embedding training has happened.
28:59
Okay then this what happens.
29:02
You can take billions of pictures of dogs and then turn them into embeddings and know that they'll all end up in the dog space, so to speak.
29:04
So Google already has done that to billions of images. Okay.
29:10
And then when you type, you know like Welsh Corgi the same embedding text also can be embedded obviously.
29:13
And so in text embedding text and picture embedding the word dog so to speak,
29:20
the word corgi Pomeranian in Alsatian German Shepherd will also be somewhere nearby okay.
29:24
So that words and pictures go together. Likewise all the different cat descriptions.
29:29
You know, kitten, kitty, kitty pool, all of that will actually go here.
29:33
And so then when you type, when you type in your word, the word will also be then embedded.
29:37
And hopefully the embedding will take your word and put it somewhere here.
29:41
Then they can do similarity search and produce results that literally say okay,
29:46
that is where you type something in image search and just shows up over and over, you know, just a plastic water bottle.
29:50
Okay. So those words will then when I hit enter.
29:55
Will then turn into an embedding, because a network already knows how to embed this properly,
29:59
and that embedding will take it to close to where previous water bottle training data was,
30:04
meaning actual water bottles, and then it correctly picks them. When I do.
30:09
This is going to happen when I click on the image tab, what I just know said happen.
30:12
That text got embedded and then the embedding was near all these bottle pictures and it correctly pulled the bottle pictures.
30:16
It's pretty highly accurate okay. So that is how image search works is the point.
30:21
Okay. So again like I said this is already there in the images. They're all been embedded right.
30:26
Embeddings vectors they mean the same thing. So when you type in a text the text also becomes embedded.

30:31
And then you get similarity search in nearest neighbors okay. You know they could give you the result at that point.
30:36
But they do one more thing. They do Reranking I talk about Reranking.
30:40
Right. So ranking means, you know, what picture should I give you? First? Second, third.
30:44
It already has that reranking means potentially altering the ranking.
30:48
Why one need to do that? Is the embedding not good enough? No, it's not about the embedding.
30:52
Embedding is great, but maybe based on your local preferences okay.
30:56
Maybe you want to see certain kinds of things but not other things. So Reranking can help take user preferences into account.
31:00
So you and I might search for the same word but you would get different results will get different results based on a past search history.
31:05
That is what Reranking is for. So once Reranking is done then yeah, it actually goes and pulls the actual images and then shows it.
31:12
Okay, great. So look at this. Then you can read all this. So how do you make a search engine in auto rank.
31:19
I'll save this for you guys to read like that. It's cool.
31:25
Yeah. So it's a ranking problem, you know, two images, similarity score and all this.
31:29
Right. So then, uh, yeah. So, like, vector stuff, you know, stuff.
31:32
And then your prompt, which is your search text will also become an embedding.
31:36
And then you get the similarity and then you have reranking. So why do you rank it in.
31:40
Because personalized ranking to the user by user specific data.
31:45
Yeah. That's the only reason for the ranking cohere.
31:49
It's a separate company and they have a Reranking API. So somebody wants to do the ranking in the cohere API.
31:53
So there's all kinds of solutions for how all this can be done. But now you understand what image search engines okay.
31:59
Here make ranks and ranks almost like trivial.
32:07
If you give it a bunch of rank URLs, meaning a bunch of URLs and some ranking, it will then give you a different list of numbers.
32:11
If your rank was one, two, three, four, it might change it to 3214.
32:17
So what you thought was number three? It should be number one. Scrambles your integers okay.
32:21
You can try all this here. Right here. The best part is it is so easy, you know, to try all this, right?
32:25

Okay. See here 123 was the initial ranking.
32:30
But then after ranking Rerank might say two should go to the top and one should go to the bottom.
32:35
So then it'll be two, three one. It's possible. Okay. That's actually what is happening.
32:39
Yeah. So you can try. It's all what you already know.
32:43
Cross-entropy loss, backpropagation, everything. Yeah. Crunch. Well, absolutely.
32:47
You know, there are so many of them. For the homework, I'm going to have you use something called quadrant for quadrant is one of them,
32:58
and then military is another one of them, and then pinecone is another one of them.
33:05
These are the top three. Pinecone is the best actually. Okay. So my homework and I might just pick one of them.
33:09
They're all Python modules. You can do a paper install and actually get them.
33:14
It'll be so much fun. But increasingly what is happening there are just a pretty good question.
33:17
Um, Postgres, which is a good old relational database from Berkeley, from UC Berkeley,
33:22
you know, so the Postgres people said, hey, we should also add vector support.
33:27
So Postgres can also now behave like a vector database, Redis, which people use as a caching, you know, database.
33:32
So Redis also has Redis vectors. Okay. So tradition Postgres there existed a long before any time any of these came along.
33:38
But now suddenly they can also become vector databases. So we call them vector stores okay.
33:46
In fact SQL like amazing SQL addition, embeddable small relational database small meaning your data across all the different platforms Oracle,
33:50
MySQL, SQL server, SQL and this will give you the smallest binary.
33:59
So you take all your data and no matter how many tables you have, it will make them all into one dot SQL add file.
34:02
So they have vector support now. So vector is now basically everywhere okay.
34:08
And I want to tell you we still have two uh topics about vectors here.
34:12
So all this assorted topics maybe I'll tell you so much I'll tell you about Transformers.
34:16
I'll tell you about many, many, many things. Okay. But yeah, the question there are vector databases.
34:20
Those would be them. There's a few more. This one called chroma midas.
34:25
All right. Chroma chroma quadrant, Malleus pinecone is all basically came out of nowhere.
34:28
Not chroma chroma, but they are pine is the very best of them.
34:33

If you go to, uh, deep learning AI, the company that I put up all the time with Andrew Nagy.
34:38
So they have courses, right? Those courses use these databases, and they have actual notebooks you can run while you watch the video.
34:44
So then you can actually get right, you know, hands on experience with things like normal voice okay.
34:50
All right. So I think I'm basically going to click on snippets.
34:55
Everything else is said okay great. So let's talk about snippets 44 slides.
34:58
But they go very quickly because the content is very light okay.
35:03
There's three kinds of snippets plain snippets, feature snippets, rich snippets, rich meaning rich media.
35:06
That just simply means video links. Okay. And these days, I mean, if you're a small kid, you don't care about the history.
35:12
Okay, I want radio and I can look at, you know, my video care speak and look at the Kung Fu Panda trailer.
35:17
In fact, what if I type that right? Suppose I type cast before I will show you rich snippets in action.
35:23
Okay. Care for it knows Kung Fu Panda and immediately overview short times all of this what's called a snippet.
35:28
It tells you who the actors are and even know the character that they play in the movie.
35:37
They talk about reviews, right? So many people and it knows, like based on the Wi-Fi, you know, where we are.
35:41
So it's picking some data nearby. Right. Well, you can just stop right there and not even look anymore.
35:46
That's very cool. Even Rotten Tomatoes score. By the way, this past weekend it crossed the $200 million revenue mark.
35:51
It cost 85 million. To make a pure profit for Dreamworks is great.
35:57
Okay, all my buddies worked on it. Okay, I didn't work on it.
36:01
Hey, you know, on April 12th, Dreamworks is releasing rereleasing Shrek two for one week.
36:05
I worked on Shrek two. My name is on there was in theaters.
36:12
It's funny, it's the ten year anniversary. I cannot believe it's been ten years. Yeah.
36:16
So this the whole snippet idea. Okay. Uh, how and where do the snippets come from?
36:20
Come in 1998. Way back.
36:25
A snippet is simply an excerpt. Believe it or not.
36:28
Are you are that little part right? If Google only showed you and basically on Google started just the URLs only, right?
36:32
You still don't know what it's about. You need to click on them to find out, right?
36:39
So the first form of snippet was a black and white text that you see there.
36:42

To pull some sentences from the URL and put it right there like a summary.
36:47
So if you don't like that black text, you can scroll and find something that you like.
36:51
Okay, so the first snippet was just simply a text snippet.
36:54
Because without it you will see. Blink blink blink blink.
36:58
And unless the link itself had nice descriptive words in it, you have no idea what the link is about.
37:01
I don't waste time going through each link. So a quadratic, for example, somebody typed what is the number of x and y intercepts.
37:05
You know. So a quadratic function will have at least two more you know at most two x intercepts y parabola.
37:12
Duh. You know at most because you know, you can have zero.
37:19
You can have one. We cannot have three okay.
37:25
Okay. So now therefore I say like that, say you don't like all of these then maybe I'll click on that one or click on I understand.
37:29
So it's just simply that in fact these are snippets.
37:35
See that's a snippet. That is also a snippet that a never.
37:39
We've been using snippets all our lives. We just did not know they were called snippets.
37:45
This is not a snippet. You know, they give you, like, all these extra real estate number in Instagram.
37:49
So if this all you need to know, right. And why I go through a bunch of URLs to find that out.
37:54
To summarize it for you, the snippets are the earliest form of search summarization.
37:58
But now it ai so-called AI. It can no more than this.
38:02
Okay. You can maybe talk to it and say you know what is poor how to dads.
38:06
Okay, well one is real. That one is a question. So hopefully will tell you all that.
38:09
But that is what a text snippet instant. Okay.
38:13
So then snippets. You know there are snippets that that are exactly.
38:17
So in the snippets there was actually dot dot dot. It's called ellipsis in grammar.
38:20
Right. That is one. It tells you they didn't make the snippet of themselves.
38:24
They literally pulled the snippet from a page itself.
38:28
So in the page in the like some sentence of the search engine, like some sentence that you wrote.
38:31
You will then pull that sentence and make it the snippet that the world will say.

38:36
Therefore, it's up to you to write good text so that it can become snippets.
38:39
Basically like advertising your page. Okay. And they won't tell you what to demonstrate.
38:42
You snippets cannot go on and on. Snippets cannot be a whole paragraph.
38:45
Maximum snippets can only have one foot six characters, sometimes only one line.
38:49
But it'll never go more than that. Okay. Yeah. And then you exactly see this.
38:53
This actually very cool. If in your page you make a meta tag, then whatever you put in the meta will be used to make the snippet.
38:57
What is your in the meta tag? See, that's a minute tag. If you know HTML, you know what I'm talking about, right?
39:06
So meta tags go in the head. There are many, many, many meta tags.
39:11
Why are they called meta tags? Why are meta tags called meta tags?
39:16
It's what? I'm sorry. One. Oh I see.
39:23
It is. But in what's. You're right. But in what sense though very specifically for like web pages.
39:27
Exactly good is not displayed because you see things like paragraph, right?
39:37
I mean, this one is an unordered list. I've got dot dot dot.
39:41
Right. But I made them into a tab using CSS. Okay. But it's actually bullet list.
39:45
These things get displayed like, you know, more stuff that's bold actually right below this like a header three.
39:48
And that's a line. Whereas meta tags are the content of meta tags are not displayed.
39:54
Exactly. So if you put a high quality text there that will become the basis for the text snippet.
39:59
So cool. Okay. But you cannot play games to find out you're doing search engine optimization.
40:05
We're putting crap there. Then they'll actually ignore your meta text.
40:10
Okay. But that is all it is. So, um, told you all that?
40:13
Yes. So your sister made a description and then. Yeah. Okay. This one is obsolete, but I can tell you.
40:18
All right. You know, the used to be this amazing thing called demos, dawg.
40:23
Wow. So the Monster Dawg is a doomed idea.
40:28
It is the same as Yahoo idea, which was when Yahoo and Google first began on the web began.
40:33
They are two different ideas. Search people. Google. People said we cannot possibly organize manually every link and place them in
40:38

astronomy and philosophy and biology will just simply will not be able to keep up.
40:45
So we'll do this thing called a search engine and make a inverted index.
40:50
Whereas Yahoo people said, well, like a librarian, you know, somebody brings a book in the library.
40:53
I know what shelf to place it. And I'll do I'll do it manually, okay. So Yahoo basically went out of business.
40:58
They're still there. But then they are not. Nobody cares about Yahoo! The most is like Yahoo but an open source implementation.
41:02
Yahoo still wanted to make money. Okay, so demands just like Yahoo!
41:09
They tried unsuccessfully to manually organize basically billions of web pages.
41:12
And finally in 2017, they gave up. Okay.
41:18
So es de mars.org.
41:22
Then. Oh, that, by the way, search completion.
41:26
I simply type D and it says dead. Oh.
41:30
See this? Well, is also another crazy fail company.
41:34
Okay? They call themselves a portal. There no reason to exist. Okay, but then they kept going.
41:37
Yes. See? Okay. Is end of life.
41:42
Uh oh. Well, you can read all this anyway. So the idea was that, you know, really get right organized.
41:45
What, what this slide is saying is if your directory was lucky enough to be listed in demos when demos was alive,
41:51
then some text from your from your page would definitely be used to make a snippet, because that means they must trust you.
41:59
Some human looked at your page and wanted to list the directories right?
42:08
It means it's worth making a snippet out of it.
42:11
Alright you guys, so then whatever that d most people put to describe your site took higher precedence than your own meta description,
42:13
because your own meta description could be bogus, could be fake news.
42:24
They could stuff it with keywords. Okay, what was the most like a third party?
42:28
An open source thing, right? That accurately describe your set. So they would prefer the most description over your own description.
42:31
But now it's a so-called moot point because no point talking about demos demos.
42:38
Is that okay? Yeah.

42:41
So then this notion of snippets, you know, I won't read every word for you.
42:45
It's a little boring. I'll just tell you this part. You see text snippet initially.
42:48
Right. And then it occurred to them, people love to look at a little image summary okay.
42:52
So let's put images up. And you saw the actors picture okay.
42:55
And then they started adding video links. The video doesn't play but mostly like a YouTube video link okay.
43:00
In fact let me do it. Okay. Suppose I type something like, you know. Always my favorite.
43:05
Suppose I see eagles in a hotel California or something, right? Let's see what happens immediately.
43:11
I'm in all sorts. By the way, I'm not in video search. The first thing that comes up with all the links, right?
43:16
I mean, look at that one. So there is now a rich media snippet because it knows I probably won't listen to it.
43:21
Lyrics. Even that is very nice. So when you know, listen to the song you want to read the words or sing along to the show you the lyrics.
43:25
Okay. Look at how many things they show you. And then there's like Spotify, you know, like Amazon Music and all that.
43:32
So all of these are ways in which snippets are being now used to augment what somebody's searching for, because that is all most people all want.
43:38
Okay. If you want, then you can get in the history of what a selection.
43:45
Meaning on the right. Some people think it's a song about [INAUDIBLE].
43:49
Okay. But it's not a song about how people think it.
43:53
All right, so then snippets keep evolving. Site links to snippets.
43:56
Yeah. You know, if there if this main link.
44:02
Right. But snippets can also have extra links in them. So they are called site links.
44:05
Entity facts are. Okay. So now that it can do named entity recognition if you type in a word like you know I don't know Elon Musk,
44:09
then suddenly it can go in Elon Musk Knowledge Graph and say born in you know like net worth.
44:18
You know, companies managed okay. That is all extracts right. We call them entity facts okay.
44:22
And then they can also do table stuff in here. I should probably zoom into some of them you know.
44:27
So then they tell you like how a stock is doing over time, you know, for example, or this one says the most dangerous sea creatures.
44:32
You know, whenever one encounter somebody made a large HTML table, the table.
44:38

Right. There's a snippet. So snippet is pulling more and more from pages and basically becoming like a mini page summary.
44:43
So if you like the summary, click on actual page. Otherwise move on. Okay.
44:49
So very obvious direction in which uh, snippets are headed.
44:52
And like why would they not do all of this. All right. So snippets are snippets of computer right.
44:56
Query time. That's also very true because they cannot possibly store from a knowledge graph all the things
45:01
in a separate model card or something like that can dynamically make it up after your query.
45:06
So many other snippets are not stored as snippets. They're being just called dynamic content generation.
45:11
Long, long, long time ago, all content generation was static HTML file.
45:16
It has some Jpeg and all links word, word, right. Some of links and you got it.
45:20
But now the page is being composed on the fly. The search engine result is making a snippet for you.
45:25
So we can do that. Yeah. And then also, you know, suppose you search for like a Reddit Reddit post, then the relevant stuff like that, right?
45:31
So in this case in our Tesla model club.com,
45:40
they know you are searching for something that could be possibly in all these, um, forum pages and how many posts.
45:43
If you search for StackOverflow Reddit, you know, same thing happens.
45:49
You can try them. Okay. So then that makes it more useful.
45:53
Like once you in this one, if you actually have a paper, then it even tells you who the paper.
45:56
So you search for paper topic it does it papers author and even how many people say that paper then scholars like that okay so researchers like that.
46:01
Again all this is to, uh, save your time from actually visiting the page.
46:09
Why not summarize it? So think of a snippet as a summary. Okay. That's really all.
46:12
Um, then this one again says, you know, a single site depending on the query.
46:17
Yes. So what kind of snippet you get, you know, can depend on even like what kind of query that you put on the site.
46:21
In other words, snippets are not site dependent by any means. Ah, okay. So any snippet can be generated for any site.
46:27
It depends on what you are asking for. So long. Snippet short snippet PR stands for people also ask that is a recommendation engine.
46:32
So if you ask something like you know how does how does browser, uh, you know, async work or something?
46:41
Okay, let's actually do that. How does JavaScript async work in a browser?

46:48
Suppose you literally ask a question and it might say people also ask something very similar,
46:54
but different words possibly see that people also ask how does JavaScript async work?
46:59
How does it work in JavaScript under the hood? They all have the words async JavaScript over and over, right?
47:05
And even in browser, but not in the same order of words that I put in.
47:11
So maybe these extra questions can also help me because, you know,
47:14
maybe I should click on one of those so they helpfully suggest they recommend other people's, uh, questions.
47:18
So in StackOverflow, that is all the time when you ask some bug that you have very similar bug
47:24
questions pop up and maybe somebody else answered your question somewhere else.
47:28
So you shouldn't just only go to the first link should click on people also ask okay PR yeah, so more of this.
47:32
See that space like super cool really even here. See that.
47:38
So StackOverflow it all like StackOverflow and even like when the post when maybe the latest post was made.
47:42
That's very useful. Right. And you can pick something like relatively new.
47:47
Okay. So you are learning like all these cool things. Okay. I'm just going to skip all of them.
47:51
So the idea is that depending on what you type, the snippets might actually vary.
47:55
And that's basically the burden is upon the user. In other words, you need to know what to type in.
48:00
Prompt engineering. This the exact same thing.
48:04
You have this innocuous looking ChatGPT prompt or Midjourney stable diffusion prompt if you don't know what to type.
48:06
If you don't know the terminology, you type something very basic.
48:12
The results will also be very basic is going to tell me basic, but if I know water type like much richer,
48:15
more keywords, more descriptive, then suddenly I get a much higher quality answer.
48:21
So in that sense, you still need to know about the domain that you're searching for is not magically going to teach you everything.
48:25
Okay? It's good news for all of us. It means the same thing can be true in software development, can be true in any field at all.
48:31
Really. You're still have to know. It just gets you there faster. So it's not going to substitute for you in okay.
48:36
All right. So let's look at the summarization this way. Again it overlaps with the latest ChatGPT attempts.
48:42
It is trying to summarize the content in many URLs and put it at the very top so that
48:47

you wouldn't have to click time going to waste time going through all the URLs.
48:53
So it's a information retrieval. Machine learning, data mining.
48:57
And now again, ChatGPT. Right. Unsupervised learning. So it creates a summary, the so-called right, by finding the most informative sentence.
49:01
There's one very big difference between this summarization that is way before ChatGPT.
49:09
This is like Alice Horowitz in our 2012 and 2022 maximum charge.
49:15
Jeopardy! Was not even a twinkle in his eye. Okay, but here the summarization works by using reusing existing sentences.
49:19
It doesn't attempt to synthesize brand new language at all. Okay, it's going to pull basically from the URL.
49:28
Now basically, if it pulls random words, it look like bad grammar.
49:34
So it doesn't do that. It will find one representative sentence, maybe abstract in a paper or something.
49:37
Conclusion, and make that to be the summary. And you like it, okay.
49:42
Because that is what the abstract is supposed to be. But the new ones, what it does though, is it is literally making up one token at a time,
49:45
one word at a time, and actually making new words that are not in the training data.
49:52
So it is actually making new language up and that it's trying to summarize that way.
49:56
So there is definitely something new going on for sure.
50:00
But the problem, as I've told you many, many, many times, is it's it's actually like, say speaking a hundred words to you.
50:03
It's doing word number one, then two, then three and four and five, one at a time.
50:10
Never, ever, ever going back to see what it already type.
50:15
It means across a 100 words that could be errors from first word to last word.
50:18
It cannot possibly tell you I'm self inconsistent.
50:23
I started telling you something and ending with something else. So you know ignore my answer.
50:27
Cannot tell you.
50:30
By the way, that's a brand new item from yesterday and I don't have the paper reference yet where people said the algorithm should do introspection,
50:32
the algorithm should do a reflection. And it's actually interesting.
50:40
Make it make it give you the 100 words I talked about, but don't give it to the user yet.
50:44
Go back and examine all of them as if it's a new prompt.

50:49
You know, like it's called. Want to start learning? Somebody give you an example and then see if that those words make sense or not.
50:53
And if they don't change it before you give it to the user. And that actually dramatically improves the answer.
50:59
So so interesting. You know that. Well, you think before you speak, uh, engage your brain before putting your mouth in gear, okay.
51:06
Don't just blurt things out. So likewise, you know that the A's actually starting to do that.
51:13
Meaning people are making you do that. That's useful. So that's not what what I'm saying though.
51:17
So what I'm saying is right now, you know, because that's not common yet.
51:22
What I'm saying it is worse than this because here they are getting existing text, but then they're at this point is making it up.
51:25
Okay. All right. Cool. So there are two ways to do it.
51:32
First one is extraction. That is what I told you. And that is what Google has been practicing all these years.
51:36
Okay. Abstraction is and see here it is.
51:40
This extractive abstraction is the new ChatGPT.
51:43
So Gemini, the challenge for all of them is to abstract in such a way that is grammatically not bad English and also factually is not biased.
51:47
It's not an easy problem. Okay, humans know because we know so much about the world, but then you want the machine to be like you.
51:56
It is not a solved problem by any means at all. Yeah I know okay, so feature snippet in feature snippet is right there at the very top.
52:02
You know, the highlight Kung Fu Panda in a short time is all that right? Because otherwise a text snippet that I showed you was below the URL.
52:11
URL text snippet URL text snippet. But now when I search for some famous thing like let's do uh, let's do Venice Beach.
52:18
Okay, you will actually see even before they show you any link about Venice Beach at the very top they showcase some snippets about Venice Beach,
52:25
right? Uh uh, here it actually listen to that.
52:32
Let me do my speech. Okay. Interesting. Okay.
52:36
Maybe on the side. You see this, right? It's not above or below any URL.
52:42
So there's actually what is called a a featured snippet. Again, they got all this from a knowledge graph.
52:46
Most likely everyone has like coordinates okay. It's pretty funny.
52:52

I mean they're all facts and nobody can deny them.
52:55
And they are in a knowledge graph and pulls it and shows it to you, whereas they still are good old fashioned snippets.
52:57
This one is a rich snippet rich media snippet because now they are starting to put up maps.
53:03
Oh my God, you guys. Wait till you hear even more.
53:09
All the data is becoming public, right? Google has no map driving around data, right?
53:13
So the augmented reality. Pretty soon you can actually put AR glasses on, apple glasses on, and you can look at parts of a map.
53:17
Okay. Or maybe pinch in front of some science.
53:25
Then suddenly the AR will take over and actually make you be inside Muscle Beach and look around and go, wow, look at all the people lifting weights.
53:28
You can be in that space as we speak again. So this is going to become like outdated pretty soon.
53:34
But this a good start though. There's so much going on. So again they're showing your photographs.
53:38
But with AR. It is actually possible to put you in the middle of the beach right there.
53:43
But all of these are now feature rich media snippets. Okay. Again, anything you see that is a photograph is a rich media in the past.
53:47
Again, as you know, 1998 Google search. No photographs at all or no related searches, you know okay.
53:54
Images, just image search. This image search, by the way, this here is the same as this image search that you get.
53:59
You're going to get a lot more. That's all. By the way Arnold Schwarzenegger actually would go there and practice okay.
54:05
I don't. Schwarzenegger is a professor at USC.
54:12
He's actually in the School of Public Policy, teaches classes once in a while. I will be back.
54:16
The Terminator is actually a precursor. Look at me.
54:22
Okay, so when I was working at Autodesk, in my speech right across from Autodesk was a building that is where Schwarzenegger had his office.
54:25
You know, I never saw him there. But you know. Okay. So what do you mean by features?
54:33
Why did you pull it out right separately? One feature snippet is a paragraph that summarized what could be in the URLs with a pretty simple text,
54:37
like a paragraph tag in HTML paragraph, or if you make a list, you know, top ten schools for computer science.
54:44
Suppose you have a page called top ten schools for Computer Science, and you make a blog post about it.
54:53
Grab your list and make a list bullet snippet.
54:57

And so now this snippet is pretty easy to read. That. Or if you make an HTML table about something in your page called,
55:01
pull your table and make that a snippet so all your HTML elements can become snippets.
55:07
And usually there are only three kinds of HTML elements in the world that are mostly paragraphs, or the bullet list or their tables.
55:13
Okay. I mean, they're obviously modest graphs in other things, right? But they don't care about all that cool.
55:18
So those are three kinds of snippets, okay. And there's even more, by the way.
55:23
You know, just go to semrush.com and then go to feature snippets. Map might be another snippet okay.
55:27
And pretty soon are data can also be snippets. Okay, this is cool.
55:32
So then of all the pages, you know that the ranking algorithm comes back with why should the pick the snippet from your page versus in another page?
55:38
Because you can make it easy for them. How would you do it?
55:44
You create a text that if there was a query, you are trying to answer, the query and voice search.
55:47
Imagine somebody asking, imagine somebody reading your page by a voice reader and the blind.
55:54
You know they cannot read properly. Some audios, you know, being like your page is being read to them.
55:59
So then you would the person be able to get something good information just by hearing your words.
56:04
If the answer is yes, that's a good snippet. So write your write your page in such a way that is audio friendly because, you know,
56:09
there are many people that actually experience the web just by hearing it, right?
56:16
And so help them. In other words, Google basically has some some rubrics.
56:19
You know, just like for the example, if you do this, you do this, you do this.
56:23
You will go high up in our list of candidates to pick the snippets from.
56:26
If you ignore those, there's no way in [INAUDIBLE] your stuff is going to become a snippet.
56:31
Likewise. What is? So if you actually in a blog post, type words like what is computational photography?
56:34
What is retrieval? Augmentation and then write some text underneath.
56:40
Most likely the text is going to become a snippet because look for the words what is okay because you took the time to explain it.
56:44
Likewise, something is something, you know. So a CI CD is a form of software development with continuous releases and continuous testing.
56:50
So when you say the word is there is a definition. And so then the like that actually pull it okay.
56:59

So you can make it easy to become a snippet basically. Also keep it pretty brief.
57:04
Don't just ramble on and on. That's obvious right? In a way.
57:08
Because say you have a long paragraph, then you become reliant on the AI to summarize a long paragraph.
57:12
That is where the problem starts. So if you make it very small, they can take the whole thing and make it a snippet.
57:18
So then keep your text like relatively short you know.
57:22
And then likewise, you know, if you want your paragraph to become a snippet then make a nice paragraph.
57:25
If you want a list and make a list and so on okay. Yeah, yeah.
57:30
So, you know, don't say things like our because it does not translate to the other person, okay.
57:33
Because when you say our it sounds like an advertisement,
57:38
but when you leave out the personal part and make it objective subjective versus objective subjective means we don't use words like i.e. we,
57:41
you know, just keep it, stick to the facts. Then then it can become a snippet.
57:48
All right. So the encoder search engine line.com is going to tell you about snippets okay.
57:52
Cool. Tldr. Okay, so, uh, yes.
57:57
Oh, this is very interesting. When I typed CRM software and suddenly you will have all these different snippets come up, right?
58:01
Okay, let's actually do it. CRM stands for Customer relationship management, by the way, a CRM software.
58:08
Or ERP enterprise resource planning centers. Consider.
58:16
Cool. So what happened? Sponsored. So very quickly.
58:21
Monday.com the people to run projects, right. Or Zoho again comes with WordPress, in which they show up even before CRM software.
58:24
Here's where it actually is. Oh my God. In fact, look at how many ads, pages and pages and pages of ads.
58:33
Wow. Because I said CRM software, the query almost is like asking, give me a list of CRM software.
58:40
Wow. Not what is CRM software? Maybe if I type what is CRM software will be the one.
58:46
Give me so many like paid advertisements. Okay my goodness.
58:52
Even here Salesforce, they still want the sales force. And they're trying to answer what is CRM software?
58:56
How can we get a simple blog post? Is the question okay?
59:01
Maybe that one. Tech target. Finally. Wow. You see, it is getting hard to get simple answers from real people like you and me.

59:04
That's the problem, actually. In other words, Google basically sold soul to the devil and then they won money from all those and advertisers.
59:14
So they're putting all those links to us. Therefore, if I want a good answer, right, you know what we can do.
59:20
What is CRM software. Read it. Oh, and suddenly that is so cool.
59:26
So they have a page called Small Business and I get my answer. Okay.
59:32
So basically people are doing their own prompt engineering by saying Screw Google.
59:35
Okay. Well just go to Reddit. So Reddit should have its own search engine, right?
59:39
But they're using Google Reddit search engine anyway. So let's move on though.
59:43
And understandable snippets okay. It's fascinating. Right? So you can get many kinds of results first.
59:46
Yeah. Yeah, I don't use it too much, but yes, it does.
59:52
Bing is very creepy in my mind because when I stare at the background picture before I log in,
1:00:02
it looks like for ten more seconds and I swear, okay, and I log in as well.
1:00:07
And suddenly Bing comes up with the the mountain that I was staring at with all kinds links to it.
1:00:12
Okay. I don't understand that. Are they attacking my. I certainly don't know.
1:00:17
Okay. Because I never said, show me this mountain in Bing, okay? I never say that.
1:00:21
Okay? I don't care about Bing, but it's creepy. Actually, no, I'm actually scared of it.
1:00:24
So click away. All right. I don't know how it works. Anyway, saying in Hollywood, we have a thing called above the fold.
1:00:28
Below the fold. Okay. Literally used to fold a piece of paper and then write down the revenues.
1:00:35
Money that they got, money that they lost. Okay.
1:00:39
So above the fold, below the form sent a search engine fold means to [INAUDIBLE] with all of the paid ad snippets in the world, right?
1:00:41
And the actual good old 1998 Google search. Where does that start?
1:00:49
The line is called the fold. So the actual standard of Google search is below the fold, and all the new stuff that they invented goes above the fold.
1:00:52
Okay. That's all. Well. Okay. Um.
1:01:00
Below the fold. Below the. In fact, the fold is also called line sometimes.
1:01:04
Okay. When you say below the line Hollywood. Below the line.
1:01:09
See this below the line. Top sheet film budget.

1:01:14
So below the line. Okay, so likewise below the fold. Okay.
1:01:17
About the fall below the fall. All right, so, you know, uh, as to say not expecting a snap is not easy.
1:01:22
And obviously simple bullets and all that they can extract.
1:01:29
Right. But what what they're trying to say is, um, sometimes word ambiguity and also uh, summarization again is not easy.
1:01:33
It's not a soft problem. So what do you want? Okay. In other words, when you type things like, you know, Tesla I don't even know what to search for.
1:01:40
Yeah. Tesla reports financial results. So what should they give you.
1:01:46
Should actually give you an NY times article. Should they give you, you know, some kind of a snippet I guess that I would summarize this,
1:01:50
this, uh, whole slide as exactly saying that the like what the user wants is not always clear.
1:01:57
Okay. So sometimes the snippet you don't really want call and see what happens.
1:02:02
Oh yeah. This actually Google this. They'll take some words you know and actually highlight them hoping that that's what you came to look for okay.
1:02:07
Doing a controller fine kind of thing. And then likewise, uh, cloud computing.
1:02:15
So anytime cloud computing, all kinds of things come up.
1:02:21
And so again, it might be, you know, the machine might not know exactly what you looking for.
1:02:24
These slides are like self explanatory but also not very useful.
1:02:28
Right. You're kind of stating the obvious. Uh, nobody, even a human, cannot.
1:02:31
If I ask you, tell me more about cloud computing. You don't know how much I know, right?
1:02:36
So you have to assume. And the machine does the same thing. So whatever it produces might not be very useful.
1:02:39
All right. So how does it generate snippets again? Um, yeah, that's actually very interesting.
1:02:45
Google actually patented so many snippet related algorithms.
1:02:51
Too many years ago on the internet first happened. Amazon tried to pretend things like one click buying and the courts actually said no one click.
1:02:56
Buying is not some secret sauce. Anybody could have come up with that.
1:03:03
Okay, patterns should be on things that are what is called non-obvious okay, so novel, useful non-obvious.
1:03:06
So one click not in order is not non-obvious. It is obvious okay Amazon got to it first.
1:03:14
So they said no to many software pattern software could not be validated.

1:03:19
But increasingly the companies bugged the patent office.
1:03:22
So now you can get all kinds of software patterns. So incredibly, um, Google snippet patents.
1:03:26
Google. Snippet patterns. What the state is trying to say is, if you want to know how the whole snippet extraction works, right?
1:03:34
When you make a patent filing called a patent application, you have to actually describe a little bit of what your invention is about.
1:03:42
It's an art to have lawyers to do it properly.
1:03:50
If you don't explain too much about what your invention is, the invention, the patent office will reject your button.
1:03:52
Sorry. Patent rejected. If you reveal too much actual pseudocode diagrams,
1:03:57
then your competition can read your patent very carefully as public document and say wow, this one thing you forgot.
1:04:02
I'll start a new company that does almost what you do, but not copy you.
1:04:08
I'll get sued, but make something slightly different and be better than you, right?
1:04:11
So you don't want to reveal too much. So. But otherwise, patents are very juicy.
1:04:15
But look at this one. Expanded snippets. Wow. So they have a patent on expensive snippets.
1:04:19
Look at this assignee Google LLC. So patents expire after about 7014 years.
1:04:24
So 20. I guess it's already expired, right? Check if you want.
1:04:31
Once a patent expires, any one of us can implement it.
1:04:34
Will not get sued. But meanwhile, that is only page one, right?
1:04:38
Look at how much you can read. See this? You can learn like this.
1:04:42
Search query based on search query. Multiple documents forming a list of results and subset of results.
1:04:45
Do you sometimes explain things like page rank? Okay, patterns are like I said, interesting because they have to tell you and I'll select right there.
1:04:51
Yeah okay. It's about a page ranking and things right.
1:04:58
Like this. Well look at this.
1:05:02
The linking based score for a search result will be based on the number of quality of links to or from the search result document.
1:05:03
That is literally page rank. Okay. Describe the page rank in one line. So you can learn so much by again like I said browsing through patterns.
1:05:10
That is the point of this slide. I have nothing more to say. Okay, cool.

1:05:16
So there's so many patterns, snippets that they make. And then here they're looking at one of them looking at almost like 20.
1:05:20
Okay look at this. It took seven years for the pattern to actually be awarded.
1:05:27
That's like so long by the way. Usually it takes about like a year. But the patent office does not like what you file.
1:05:30
The one out rejected does say return for comments that come back with version 2.0 that tried for seven years and finally got it.
1:05:36
Wow. All right. So this one is snippets, you know, snippets generated based.
1:05:44
Oh yes. Look at this type of the query or the location of the query of the terms in the document.
1:05:49
You know, so if the query that the user put in is at the bottom of the document, maybe then that won't make for a great snippet.
1:05:56
I suppose. Maybe the top of the document they're using somehow where the query is in relation to
1:06:03
a document like words to decide if the document should create like a snippet or not,
1:06:08
that is described here. Okay. It's one patent.
1:06:14
Well, anytime you think of something very cool, I guess lawyers are gonna abandon the reason why the patent is because your computation in this case,
1:06:16
Bing, actually might do something that violates your pattern.
1:06:25
You know, because if you patent how snippets are made, Google Bing also has to make snippets, right?
1:06:29
So then then Google will say to Microsoft, you're basically using a patented algorithm to pass royalties.
1:06:34
So Microsoft being Microsoft also has a bunch of other patents where they say the bulletin is formatted like a certain way of writing in that,
1:06:41
and they tell Google, look, your search results look like ours. So we're going to show you basically the each they cancel each other okay.
1:06:48
Many companies do that. I have patents are up patents like playing a chess game.
1:06:54
You got you got five of my pieces here. I'll take five of your business. Let's go home.
1:06:59
Okay. Game over. Just tell me.
1:07:03
Okay. So location based rules like, not location like Los Angeles, but times in the page.
1:07:06
Wow. From the start or the end. So you have the start of the page.
1:07:12
End of the page. Right.
1:07:16
If there are some words there, like at the very top can become a snippet or at the very bottom can become a snippet or somewhere in the middle.

1:07:17
How far is the distance between the top and the bottom? What happened in the middle?
1:07:23
So if the distance is too long, then maybe it won't be a snippet. So I think all of that is somehow in the pattern.
1:07:28
If you have things at the end, you can use it to summarize, because normally when you write a paper or something,
1:07:34
right, the conclusion is going to summarize your whole paper. So that snippet will be naturally like a summarization snippet.
1:07:39
So I like so many, you know, I mean this is like weird right? NLP see this number of bold italicized words, even punctuation characters.
1:07:45
The use. Basically it is called ad hoc rules,
1:07:53
basically made up rules by trial and error that gives them good snippets and then the hard code alter and the search engine.
1:07:55
So there's no rhyme or reason. If you say, why are they doing this? Well, you know that.
1:08:03
Infinite time to find out. It works for them. I cannot give you a good reason.
1:08:07
Likewise, if your words are too short or you have again,
1:08:11
way too much punctuation or something right your way too many italicize bold words, then you will not become a snippet.
1:08:14
You know there's some rule of thumb, okay? Rules of thumb. We'll take a break at 630.
1:08:20
Okay? Okay, so this 8,000,000th patent, you know, about how do you do snippets.
1:08:25
You know, again see all this. Yeah. So identify paragraphs.
1:08:29
First of all they include quite this easy right. Mechanical.
1:08:33
Take the query term and go into document. Find all the paragraphs that contain the query term.
1:08:36
But will they become a snippet or not. For that you have to actually do like all of this.
1:08:40
Uh, is the the query term at the beginning of the document then score that.
1:08:44
Well, yeah. I said there a threshold okay. Some kind of a number of words.
1:08:50
So if any paragraph contains the query term, but if the paragraph only 30 words long they have a threshold called 50.
1:08:53
I'll only look at paragraphs that are 50 words or more. Your paragraph is only 30 words.
1:08:59
So even though it has a query term the user wants, your paragraph is gonna give you a score of zero because it did not meet the threshold.
1:09:04
Okay. You failed basically. Cool. So likewise. Kate.
1:09:10
Oh, that's so crazy. I'm not going to read all this. Okay. So two ad hoc.

1:09:14
Okay. Some kind of Kate paragraph from the start to make some kind of a numerical value.
1:09:17
Okay. For every single paragraph that contains your, uh, query term and the highest value paragraph wins, I guess.
1:09:22
Okay. Yeah, yeah, it's too ad hoc. Okay.
1:09:29
So on purpose, I'm like, no, I have better things to tell you. Okay.
1:09:32
That was, uh, 8145617.
1:09:36
This one is 863. One more time has gone by the pattern number.
1:09:40
The pattern number variable monotonically increases. Okay.
1:09:44
This means ever since the US pattern of I started to 8,600,000 patterns before this one, right now I think one might be up to 11 million or something.
1:09:47
In fact, wouldn't that be cool to actually Google it? What is the.
1:09:55
What is the latest U.S. Patent and Trademark Office PTO patent number?
1:10:00
How would it know? Right. But it might give you, like, a ballpark estimate. Wow, I was right.
1:10:09
11 million. Oh my God, 11 million things have been invented just in the U.S.
1:10:14
Okay, I bet it's a worldwide. It's China, Japan, India. All right. Cool.
1:10:18
Right. 11 million patents. Wow. And this was when 2021.
1:10:21
So now it's obviously one market. By some measures.
1:10:26
China and Japan. Suddenly are in a race with the US for a number of patents being filed every year.
1:10:30
Naturally, going up in this country is going to taper off. Interesting.
1:10:36
All right, so that particular pattern is personalized snippet.
1:10:42
Whoa. So now, like I told you, the same search term you can search I can search.
1:10:45
We might actually get two different feature snippets because it is based on your search history,
1:10:50
which is my search history because from the search history, they know what you like and know what I like.
1:10:55
They can customize a snippet based on that. Okay. So it's not the same snippet.
1:11:00
Search for all of us at all. So that is what this does. Look at this user profile.
1:11:03
Whoa. Okay. With a note of about me. Right. In my case, it might say, you know, professor at USC, computer science.
1:11:08
Then whatever I search might be matched with that, and I will get more CSS oriented results.
1:11:14
And if you like scuba diving and that's in your profile, then it will take the same search term and slant it more towards

that okay.
1:11:18
These are fascinating experiments that two of you can get together and try to search for the
1:11:24
same term and then compare in the same search engine while you're getting different results.
1:11:28
Okay. Somewhat different. Okay. So otherwise just keep all this.
1:11:32
Um. So what is this one? Okay. Yeah. So.
1:11:37
Yeah. Okay. So how many paragraphs? Snippets. Okay. In in a bunch of.
1:11:41
Suppose you do a bunch of search. Okay.
1:11:44
And you look at your count how many types of, um, the three types, how much of each, like how many paragraphs, snippets.
1:11:46
How many list, how many table. You can see the clear difference, right? 70% of snippets are paragraphs.
1:11:52
Because people easily can read a whole paragraph, they can grok it, right? A table is nice, but table is.
1:11:57
To summarize, there's only just in keywords and some tables. Some columns.
1:12:02
Right. Um, lists in between. But yeah that's clearly this.
1:12:05
So interesting table is only 1%. Um okay.
1:12:09
And then also the ranking position. Right. So like where does a snippet actually show up in relation to your whole, um, you know,
1:12:13
placement of all the results, you know, so how many show up in position number zero.
1:12:20
Not that much. Many of them show up after position number one.
1:12:25
Turns out there's a monotonically decreasing number here. Right. So interesting.
1:12:28
They go here and then it jumps to zero over there. All right you guys.
1:12:32
So again what. This is so cool. Pay attention to that okay.
1:12:36
Might be an exam hint or something. What kind of snippets do we see when we go search for things.
1:12:39
You see all these right. You most likely see images. You see sites.
1:12:44
You also see brand names. Those are like sponsored snippets. You see Wikipedia entries and then we see ads.
1:12:48
Ads all like brands. They're pretty similar. You see named entities, maps most certainly.
1:12:53
You know, one at a muscle beach showed me on Muscle Beach was and then it talks about city where something is in.
1:12:58
Hmm. What if I just say the dorm or something, right. If I say the dome or the sphere, actually, then it should hopefully say Las Vegas, right?
1:13:03

Let's see if that is true. Yeah, it says Arena in Paradise, Nevada.
1:13:12
Cool. So that is the place part of this. Okay. Or even this one actually gives you that.
1:13:17
So snippets are cool and then news. Yeah. If I type again I'm just a search word.
1:13:22
Right. Suppose I say bridge collapse. I don't even need to say Baltimore.
1:13:27
All I typed was buried and I would not use this computer in my search.
1:13:31
Okay. It is all the other people in the world searching. So today, this morning, there was a bridge collapse.
1:13:35
None occur in the eastern U.S. So bridge collapse.
1:13:40
As soon as I say that. See this top stories. It's giving me news links.
1:13:43
Okay. Because it's not the time for me to go read about how bridge bridges collapse.
1:13:47
Okay. I think I want the news. So they put up the news stories in the very top, right?
1:13:50
It is so cool. Baltimore Sun, the city where actually it happened.
1:13:54
NBC news, whatever NY times. So cool. And then at the end, I mean still news.
1:13:58
Look all news news, news news. So smart. All news.
1:14:03
Oh, okay. I mean, it just goes pages and pages of news, right?
1:14:07
And YouTube videos. Wow. Finally, things like History.com, you know, devastating bridge collapses.
1:14:11
So that's actually super cool that it knew that I'm looking for a search snippet.
1:14:17
So it's giving me a new snippet rather than giving me all these news links.
1:14:22
Okay. Fascinating, huh?
1:14:25
Okay. Yes sir. People also asked. There is a form of snippets also, right, because they are highlighted.
1:14:30
So when you ask for something, they take your words. That does a similarity search by similarity search like this is used for so much.
1:14:36
All the StackOverflow, Reddit Qualcomm questions are always similarly.
1:14:43
The questions that we will ask have all been vectorized.
1:14:48
So similar queries, similar vectors will end up in a similar location.
1:14:52
So when you search, your query will also end up in a similar location.
1:14:55
That is what they can do. People also ask look a nearest neighbor search okay.
1:14:59
That is one of the most important things ever. Like I told you, there was an exam question.
1:15:04

What is the most fundamental measure in information retrieval? And so a similarity.
1:15:08
You guys wrote some other things. You know that might be somewhat okay, but the what the thing I was looking for was similarity.
1:15:13
It's all based on similarity. In inverted indexing it is literally keyword.
1:15:18
You know, um, so similarity but slowly with so-called I it is more like low similarity.
1:15:22
The word that I search for might not be in the result that I'm getting, but it's still relevant okay.
1:15:28
This is more soft search. Okay then guess what. Cosine similarity.
1:15:33
We take dot product angles. Think of the angle between two vectors is very small.
1:15:38
The dot product angle is zero. Cost of zero is one. So pretty high.
1:15:42
Uh cosine similarity right. Okay.
1:15:46
So people actually looked at cosine similarity I think over on Netflix to figure out somebody did an and recently and said here's cosine similarity.
1:15:49
All that it's cut out to be. Can it have problems. And the answer is yes.
1:15:57
Cosine similarity can sometimes give you absolutely bad results.
1:16:01
So I shouldn't think cosine similarity is the be all end all. So I'll find the link for that tonight and put it on extras okay.
1:16:05
So please read done. I was very fascinated because I secretly I want to tell you, even though I teach this all the time.
1:16:11
Right. And I think about it and it appears to be pretty cool.
1:16:17
I've always wondered if Euclidean distance, you know, these kinds of distance calculation, angle variation are really everything.
1:16:19
Meaning, you know, is that the ultimate measure of, you know, similarity? I've always had my doubts.
1:16:26
And somebody actually did like a big study and said, wow, look at all these examples where it's bad, you know, um, so they propose alternatives.
1:16:30
Okay. Oh, there's something called manifold search.
1:16:37
Okay, so now we're off on a tangent for a minute. Manifold search.
1:16:40
Right. What is called manifold embedding. This is a very different way of doing it.
1:16:44
In manifold embedding. You actually take the structure of the data and then you start to find similarity, not some x y z axis okay.
1:16:49
And use it for images and audio. Okay. Scenario x y z.
1:16:57
So manifold embedding is incredibly amazing. So you guys can go and study that that rather than the Euclidean distance you can

1:17:00
do similarity based on some some paper that you've bent in a weird shape okay.
1:17:08
That's called the manifold.
1:17:12
See here the distance between this and this might actually be closer than the distance between like you and on the top okay.
1:17:13
Meaning after you fold something, suddenly things that are far apart in Euclidean distance might suddenly become closer in manifold distance.
1:17:21
Do you see? So manifold means on the surface, on some complicated surface, not even simple surface.
1:17:29
So those kinds of similarities, in my opinion, are a lot better, actually.
1:17:34
Oh, one more amazing little rabbit hole. It turns out, and almost all similarity.
1:17:37
Why are you and so almost 100% of all similarity, even Euclidean distance you in probability the so-called KL divergence.
1:17:44
How are two curve's probability distributions similar to each other? Vector similarity.
1:17:53
Distance similarity. Any similarity? Okay. Anything at all that I ever said.
1:17:58
Is equivalent to the following. Moving piles of sand around.
1:18:03
What? You know, so when piles of sand are very close to each other.
1:18:08
It's actually very easy to swap them when I have one pile of sand there.
1:18:13
One more pile of sand over here. It's not that easy to move them around, right?
1:18:17
So in that sense, that is not highly similar. These are highly similar.
1:18:20
What? Okay. Transport theory. Transport means transporting materials.
1:18:24
Transport theory. Okay. Well, it come up.
1:18:29
I have no idea. Transport theory.
1:18:34
I'm going to say a machine learning similarity.
1:18:38
Machine learning similarity matrix okay. Similarity matrix.
1:18:42
We have to see. Uh, yeah, exactly.
1:18:48
Okay, so 2020. Very cool. Yeah. So now it's getting more and more recent right.
1:18:54
Maybe I'll just pick and pick that one. Optimaltransport you actually have to look at this.
1:18:58
Okay. So Optimaltransport, this idea of this whole sand pile movement is very interesting.
1:19:03
Look at the people. Mathematics and statistics. Okay? At some point.

1:19:08
Okay. So all of these dissimilarity measures see that there's so many similarity measures, so many of this on this.
1:19:13
Okay. That all it turns out related to material transport.
1:19:20
Right. And it has applications in machine learning. Oh my god.
1:19:24
Okay. So we have to come back to snippets people okay.
1:19:28
So people also ask I understand people also ask snippet is growing at a much higher rate than standard snippets, meaning most people like this.
1:19:31
Google is sharing more and more of those over time. Okay. Makes sense. Okay.
1:19:39
Recommendation engines okay. Rich snippets, so suddenly things become more visual.
1:19:42
It means it's mostly a video image, like a map, or maybe a still like a photograph, you know, or maybe a video link.
1:19:48
Sometimes audio. I should maybe say Taylor Swift. Okay.
1:19:55
Okay, let's look at that. Hopefully it'll give me like a Spotify playlist.
1:19:58
Actually I look at something very different. I'll say Nandini Shankar.
1:20:02
I'm going to say Tarana. That's what happens.
1:20:08
Oh, cool. Now go on. You see it?
1:20:13
Made a video. I was hoping for, like, a Spotify playlist or something.
1:20:16
Right? I'm not sure. Okay. But yeah so I didn't there's no there's no work of music there.
1:20:20
Right.
1:20:26
But incredibly it actually made I'm going to I'm usually for one minute there's one simple hearing hack anyone can use to interrupt music links.
1:20:27
Okay, not that I'm actually looking for music. Did you? Hey, check this out. The people that are going to show up and play.
1:20:35
The sisters are India's foremost violinists in all of India.
1:20:45
1 billion people. Who is at the top? Those two of them.
1:20:49
How do I get back to. The mom and dad.
1:20:54
Music professor. Grandma. A pioneer.
1:20:59
For, right? Fireworks.
1:21:10
Griffin. To make it sound easy, right?
1:21:20
Years and years and years of practice. A violinist and use this instrument to screw up one little micro, one millimeter off

the console.
1:21:27
Look just how horrible. They're perfect.
1:21:36
Well. So then I was hoping for like a see imagine a Spotify playlist shot up right?
1:21:41
Or something, right? That would be an example of that becoming like a rich audio snippet.
1:21:46
So I want to move on. Okay, so Rich snippet is simply all these newer forms of summarization.
1:21:49
Pretty soon I told you they might detect that you're wearing like an AR glasses,
1:21:54
or they might know that you're going to the link in your AR glasses and suddenly they start showing your location data.
1:21:58
You actually feel like you're in that location in 3D. All right. And so that is possible by more neural network.
1:22:04
Um, magic. It's called a nerf nerf nerf.
1:22:10
So now stands for neural radiance field. General radiance field means random people can take pictures of this very room over many years.
1:22:14
Okay. And then you can take all of those pictures and then make a continuous representation of this room,
1:22:22
and then make new camera angles and fly anybody through this whole space.
1:22:28
So it's not calibrated in any sense at all.
1:22:32
I didn't take all my pictures with one phone, one lighting condition, any phone, any cropping, any lighting condition, any angle.
1:22:34
You can switch them all together. Machine learning magic and make like a quadcopter a continuous move from any angle to any camera.
1:22:40
Basically. Okay. That is called nerf.
1:22:48
So therefore, because so much data is available to all of this, we can now make Nerf renderings of basically almost any place on earth.
1:22:49
So therefore this will all become very old fashioned soon okay. You will know when that happens.
1:22:56
Okay. So rich snippet example snippets. So when you type somebody famous not me, then suddenly will show you the LinkedIn profile.
1:23:01
I'll show you that Facebook. You know what? All right. Yeah. I think I heard that I'm not trying to, you know, scoop last minute.
1:23:08
Yeah, I think I heard that where, um, there was certainly some kind of seedy level, you know, this whole Nerf stuff going on.
1:23:31
Yeah. Yeah, it's actually very cool. So then Google would obviously use that.
1:23:38
You know, if Nvidia has an API, Google maps exist actually that they're all working together.
1:23:41

You know, Google has 3D data sets for almost any place in the world.
1:23:45
Actually, no, that's not even that, though. There was something else I'm trying to think of is another site.
1:23:48
You guys. It might be called something like locality dot I,
1:23:53
I'm just making it up okay or perplexity later er where we or encouraged to upload
1:23:57
any pictures we took of any place and they do a crowdsourcing and turned all,
1:24:02
all that into Azure data. I don't remember what that is right now, but I look at it afterwards anyway.
1:24:06
So people search. Okay. This comes from obviously named entity recognition.
1:24:11
If it knows that you know somebody is famous, right, then it'll start giving them all kinds of things that they wrote,
1:24:15
you know, start giving you the Facebook information, linked information okay.
1:24:19
That is a people search. Likewise events. Okay. Suppose I type Woodstock or something.
1:24:23
You know, Lollapalooza, you know, electric, you know, Daisy Carnival first of all.
1:24:27
Then it knows it's an event. So it'll start. In fact, let me do that.
1:24:32
Okay. What if I say EDC? What if we just say EDC?
1:24:35
Wow. So it instantly knows that it's an event, right? So I start telling you, all of these founders in attendance, all that.
1:24:40
So cool, right? It is. So I just knew from that acronym that's what ADC stands for.
1:24:47
So it made an event snippet like right here all the events in the brand.
1:24:51
So it's getting more and more intelligent. You know based on like all the words that are we research for.
1:24:55
Okay. So why rich snippets should take a break. Okay.
1:24:59
Because you know here you can obviously you know make.
1:25:03
So webmaster meaning people like us that make up pages. We can then help the search engine summarize.
1:25:06
You know that's good. And then just for the user when somebody's searching,
1:25:11
they can just simply pick the page that they want to click on because every page
1:25:15
is basically giving you a little advertisement but the page and then yeah.
1:25:19
So then from a webmasters point of view, if your snippets are very good,
1:25:23
people are going to visit you, then you will have like more high traffic traffic.
1:25:26
And if you put ads then hopefully somebody will click on there and then everybody can make money.
1:25:29
Okay. This just this whole slide is like why are snippets cool okay.

1:25:33
Yeah. You know. Yeah. And then yeah how do you do snippets.
1:25:36
You can very easily make a meta tag. It is not too much.
1:25:40
It doesn't change the appearance of the page because snippets on the the matter content is not rendered right.
1:25:43
So you don't have to go out of your way to change how your page looks,
1:25:49
but you can make it more useful for the search engine and eventually for the end user.
1:25:51
Um, so this one is again more okay. Yeah, schema.org is actually very cool.
1:25:56
I'll just tell you briefly about also. Okay.
1:26:02
You know, this notion of standardization is always an important idea because have Bing uses their own format meaning Json tags,
1:26:04
keys and values or XML tags, even with the one vocabulary they call the place lowercase place.
1:26:13
Okay, if Google wants to do the same thing, they also want to make snippets, but they call place with uppercase SP and then lowercase LSI.
1:26:20
The words are not interchangeable, right? Just not a standard. So we cannot basically swap snippets in the future if we want.
1:26:27
Okay. So to avoid that you know, things like schema.org you know so they all are microdata format.
1:26:33
It is basically uh, attempts by these, you know, competing companies to come together and say, can we agree on a common set of keywords,
1:26:39
common set of keys, so that all the values can then be in terms of those keys, so that then we can very easily like swap, mix and match.
1:26:46
We have repositories for them. That is all it is. So one of the such items was called schema.org.
1:26:53
I'll even show you schema.org schema. So schema means basically table definition right.
1:27:00
So I show you schema.org. All right. Uh schema definition.
1:27:04
Schema definition. Like how would you specify the word birthday.
1:27:09
What is a birth date. What is a birthday. Are you guys.
1:27:13
In the back. Okay. Select this semantic vocabulary of tags or microdata.
1:27:17
Tags are tags are tags are tags okay. Select this name author.
1:27:25
So authors should always be called lowercase author. That is all okay.
1:27:29
So when you use common like aggregate reading so how do you call Yelp rating.
1:27:33
You know Rotten Tomatoes. We used to avoid aggregate rating with that format with that formula with the spelling

rather.
1:27:37
So when we use common words for tags okay, then the tags become more meaningful.
1:27:43
That is all. So but there's many kinds unfortunately.
1:27:48
Okay. So the standards you would think there is only one set of standards.
1:27:51
But you can see there are three sets of competing standards okay.
1:27:55
They all want to be the standard. So the joke is there's no standards for standardization okay.
1:27:58
Come on. It's pretty crazy. So schema.org is not the same as metadata.
1:28:03
So make sure to use schema.org schema right I can even tell you micro data microdata schema definition.
1:28:07
Now you will get something completely different. Here.
1:28:16
Yeah, so I should have guessed Microsoft or something. So their vocabulary, like whatever they say, what organization place so to speak,
1:28:25
might not be compatible with other, you know, set of tags that I showed you, right.
1:28:34
That's also a problem because then some companies will say, well, like this,
1:28:37
some other companies will say we like the competing format, then there's no standard anymore.
1:28:40
You go back to square one, it's basically B.S. Okay. Yeah. So we have that going on.
1:28:44
Okay. Uh, 634 maybe I'll do like one more, uh, slide.
1:28:49
Okay. This is very cool, right? Json ld actually, that's pretty cool.
1:28:54
It's called Json link data. I'll tell you a little bit. Just generally take a break okay.
1:28:59
And you can in a page have all the tags. And depending on what search engine comes and looks, you can serve all of them with Json-ld.
1:29:03
Suppose you make a recipe. A banana cake recipe.
1:29:10
I mean, if you use certain approved tags, that additional standard specifies like cooking time, number of servings, you know calorie content, right?
1:29:14
Then what Google will do is take your recipe from your page, from your blog, maybe, and beautifully format it visually so they can make it appealing.
1:29:25
They can make the picture of the banana cake and the cooking time, the ingredients, right, and format it beautifully.
1:29:33
That is called Json-ld. So as long as you use Json-ld, Google will then take advantage of that and make a very cool little recipe.
1:29:38
Okay, so I'm going to show that. And maybe that's a break. You can mix them in the same page.
1:29:45

When you make a page you can use the Json schema schema or Json-ld.
1:29:49
Just throw all of them in the same page. And that's totally okay to do.
1:29:54
There are obviously differences between them, right. Just about oh well okay.
1:29:58
So regardless of what all of those are, you can put them in the same page because in your page.
1:30:02
Right. Json-ld recipe example.
1:30:06
Okay. Oh.
1:30:12
Recipes. Schema markup. Check that out.
1:30:16
Uh structured data uh guidelines, which does construction examples.
1:30:21
Oh, look at that. Okay, so say you want to make a recipe for, you know, pina colada nonalcoholic.
1:30:27
So I type this recipe because it's a recipe type. And then image you need to say image and give it a bunch of images.
1:30:33
Then they'll put that at the top of the recipe okay. And then author. So host the recipe.
1:30:39
Date. Published recipe. Cuisine. Is it American Chinese and Indian.
1:30:43
Right. Prep time. Cook time. Total time. How many hours?
1:30:46
How many minutes I guess, or preparation time, I think. I don't know what it stands for.
1:30:50
Okay. Keywords nonalcoholic recipe. Yield. How many servings does it make, right?
1:30:54
Nutrition. How many calories? I told you, if you use the format, then the bonus price that you get is that when the search engine picks it up,
1:30:58
they'll format it beautifully for you. So then what does it actually look like if you do all this?
1:31:06
Guidelines recipe, author keywords. Worst actual example okay.
1:31:11
Mhm. I'll just go to Json-ld recipe and then just go image search.
1:31:17
Okay. So it's more like it's probably like uh like that.
1:31:23
That's exactly what I mean. That. So maybe this was done.
1:31:28
So this is some blog, right? Some cooking blog. You know, Eleanor Davis.
1:31:32
But see, in other words, they are not, you know, uh, tech people like you and me, the content creators.
1:31:35
So she supplied it like an example and ingredients and, you know, keep going on.
1:31:41
Right. Therefore,
1:31:44
the point of all of that is the retrieval can become nice for the end user if you follow certain guidelines like use Json-ld tags in this example.

1:31:45
Okay. Let's do a ten minute break from 637 to 640.
1:31:54
Sorry. Ten minutes. Okay. Cool.
1:31:57
Woo hoo! Rock! Jason Lee, ladies and gentlemen.
1:32:02
Yes, sir. Hey, this.
1:32:12
Also this. Why not? Right. Check this out.
1:32:23
Oh. I'm going to save this for a little bit later, but check this out.
1:32:29
Right. Oh. If you've ever tried to play guitar but your fingers wouldn't cooperate.
1:32:35
It's not your fingers fault. In fact, it doesn't have anything to do with your fingers at all.
1:32:43
Or your age. Backwards. Check this out. All right.
1:32:48
You fill up my cell. And say.
1:32:59
And like a knight in the forest, collected some like mountain spray.
1:33:03
I'm like a walk in the.
1:33:12
Like a storm in the desert.
1:33:19
Like a sleepy blue ocean. You fill up my cylinders.
1:33:24
Come help me again. Um, let me tell you.
1:33:34
Let me give my life to you. Let me drown in your lap.
1:33:44
After. Let me die in your.
1:33:50
Let me lay down beside. On you.
1:33:58
Let me always be with you. You've hung up my son before.
1:34:03
You can love me again.
1:34:12
Okay. Well, finish it afterwards.
1:34:21
Yeah. Hey, you should come and sing with me. Why not?
1:34:28
Right. Grow John Denver. Hey, you know, I keep offering this.
1:34:32
Okay? There's still four more weeks left. Okay. For the instruction.
1:34:36
And the rest of today, anyone can come pull up anything on YouTube and sing, you know, or dance if you want.
1:34:40
Okay? I can juggle. Anybody. Please do it. Okay.
1:34:47

If you don't do it, I'll keep doing it. Or you can join me. Okay.
1:34:50
So. Snippets. The whole schema. I showed you schema examples.
1:34:54
We can skip some of this detail okay. There are basically three competing standards okay.
1:34:58
One is called schema.org. The other one is Json link data.
1:35:04
And the third one is simply call microdata. There's lots of chatting going on.
1:35:07
I'm going to hold up my little, uh, you know, Robert's Rules of Order.
1:35:12
Stick one at a time, please. I'm going to go. And I'm going to do attendance very soon.
1:35:16
So if you're chatting, you might miss your name. Five points docked.
1:35:22
Oh my God. Okay. So please listen. Okay. I was going to tell you that.
1:35:26
Yeah. So in the end it comes down to person, place or thing okay. Everything in the universe.
1:35:30
And you play 20 questions. Is there a person, place or thing? Lots of things.
1:35:34
A person place or things. But what actual word would you use?
1:35:37
The word person is uppercase P versus at person.
1:35:41
Are called tags. They're very specific tags that some organizations come up with.
1:35:44
The idea would be if all of us use the same tags, they're easy to parse and they're easy to format, that's all.
1:35:48
But unfortunately, there are three competing standards. And so, you know, people that put up content would need to pick between them.
1:35:55
Or if you want, you can actually make all the three tags interchange them.
1:36:01
And then, you know, some search engine or the other would actually catch it. Okay. That is what the summary for all of this is.
1:36:05
So then the summary would be what? Oh.
1:36:09
You know, it's very funny. The scooter in the back, right? It looked like somebody had raised the hand.
1:36:14
That is so funny to me. A student sitting next if I took a picture.
1:36:18
It looks like you have a mechanical hand that's going up like that. Resist.
1:36:21
No. Yeah. Look at it. Look behind. See that?
1:36:25
You know she's sitting next to it, right? Yeah. You. And then. And then you're like, see, this is what it looks like.
1:36:29
It looks like that. We're good. That's good. Sorry I raised your hand.
1:36:34
Um, yeah. Sorry. The whole idea would be, you know. Right. You can describe all of these things and using structured,

it's called structured text.
1:36:39
Then if you use structured text, it's what. It's a form of AI.
1:36:47
They AI is very cheap because a machine does not have to wonder is it a person?
1:36:50
Because your key says that person. It's got to be a person. So you are basically making the keys meaningful.
1:36:55
So the values become automatically meaningful. All right.
1:37:00
So when we inject our intelligence into all of that review ratings, if you use the accepted text,
1:37:03
they'll know it's a movie rating and they'll turn that into number of stars. Okay.
1:37:08
If you use your own words, then they might not know what it is because who knows what.
1:37:11
What do you mean that is all. So these become useful for snippet generation and then.
1:37:14
Okay. All right. So again more about rich numbers. So same idea right.
1:37:18
Microdata microformats. You know they all mean the same thing. So how do you actually use microdata.
1:37:23
You say things like item property. Now say like here if you say things like class equal to C okay.
1:37:27
So you know in HTML there is something called class okay.
1:37:34
You can make up classes for all the CSS you know entities right.
1:37:37
Same for the HTML entities actually. So you can use the class mechanism like when you say class called a title.
1:37:42
And then you say the word senior editor and you put it in a span. Span is also an HTML tag.
1:37:49
You know, not a paragraph span can be anywhere. Then you can pretty much like abuse this class format.
1:37:53
Because classes are supposed to be for visual purposes. Class is supposed to be make this big 20 point font okay.
1:38:00
That is what classes are for. But we can hijack the class mechanism and then put in keywords that we want like addresses ADR titles.
1:38:07
Ideally then the system will know that that's actually somebody's job title.
1:38:15
It knows that this is somebody's zip code. Say the word postal code.
1:38:19
So it's a form of cheating to form of Trojan horse, meaning the class mechanism in HTML was not invented for data purposes at all.
1:38:22
It was invented for visual purposes. But then we are actually hijacking it to do like what we want.
1:38:31
Okay. So interesting. Like, you know, in Java doc, write the Java, uh, comment format is simply I can put in my code.
1:38:35
Whatever. But if I add an extra star and I say things like an image or something,
1:38:43
right then suddenly that can become the class name for the class that I'm writing.

1:38:48
In other words, that extra star turns it into an automatic document generation system.
1:38:53
It is called JS doc, right? You probably know all this.
1:38:57
So whenever you're writing any function, any module, any class, then decorated with these extra tags,
1:39:00
then a parser can go through automatically and pull all these keywords out and turn them into beautiful links.
1:39:08
It's a lot like this okay. So structured data if I use it who knows.
1:39:13
Horse use Javadoc. Somebody is going to put up their hand. Yeah a few of you.
1:39:17
That's a Java doc is super cool. I'm just going to show you one example.
1:39:22
I'm going to say Java Doc example okay. So look at this Java example then everyone.
1:39:25
It is also an example of what we're talking about which is structured text.
1:39:30
In other words, if you do something cool, if you do something that is uh, expected of you,
1:39:34
then in turn you will be rewarded because whatever you type will suddenly become automatic documentation.
1:39:39
See this slash star? Star in extra star turns it into a Java doc.
1:39:44
If I remove this extra star, the whole thing won't work. There's a star here as well, but there's a new line character.
1:39:49
Okay, so only these belong together. She returns an image object and returns immediately at parameter.
1:39:54
At return at C. At C means link. So then you write your code as usual.
1:40:00
But that will become automatic documentation okay. I can show you the my API format.
1:40:06
So you look at this if I say Maya API documentation, I used to use Maya Dreamworks now for so many years.
1:40:11
Right. So then they would do exactly this kind of decoration like you said.
1:40:17
Maya classes. Class list. Suppose you say am 3D view.
1:40:22
This is an OpenGL viewer. You look at that right. All of these.
1:40:26
So these are all done. In other words all of this is automatically generated.
1:40:29
Believe it or not, nobody went and typed all this okay.
1:40:32
When they wrote code, it structured it in a certain way that we can jump from, you know, link to link to link.
1:40:35
Okay. So very cool. So our example of snippets is very similar.
1:40:40
If we use those classical tools and very specific keywords that go in class equal to,
1:40:44
such as ADR photo twin is a classical photo that's going to be used as a picture on top of your recipe.
1:40:49

What would run through your school? All right. So we called for our title or, you know, idea or locality.
1:40:55
I don't even know what we call this business card movie. I don't know, no, actually we called it turns out.
1:41:01
Yeah, it's like a virtual business card. So any classes can be telescoped, right?
1:41:06
They can be nested in order to make an order, they call V card X somebody virtual business card.
1:41:11
In there you make a picture, take a picture of them and call it classical the photo and then finish first name.
1:41:16
Oh, interesting. It says Bob Smith. No answer. The title senior editor act and then organization Acme reviews.
1:41:23
Right. This can be like Fandango and then address can be broken up into street address locality zip code or TSA postal code.
1:41:31
If you then put your data in that format,
1:41:38
then they can then they meaning the search engine can present this to the world in a nice virtual card like a little, you know, business card format.
1:41:41
If you screw up on those keywords and leave all that out, then it's going to plaintext formatted right, not null, not going to look good.
1:41:48
That is all. So this format is very fascinating because also social media tag formats that I want you to look up as an exercise.
1:41:54
Okay. Okay. So that is up to you. Why do I say I have like a LinkedIn account in our Instagram account.
1:42:02
All right. I can do the same thing. I can have a record where I can say, this is my LinkedIn URL, this is my, you know, Twitter URL.
1:42:07
But except I lose certain of those tags that I want to look up, then what will happen is when you search for my name, magically,
1:42:16
the meta Facebook tag will have the nice meta icon next to it,
1:42:23
the Instagram tag where I have the cool Instagram icon because it knows to convert all those tags to icons.
1:42:27
Okay, so those are like rich media icons, so please look it up.
1:42:32
It uses a tag format, but I won't tell you what it is okay. I'm going to have you do some work.
1:42:35
I'll write some more, which can be described very similarly like that as well.
1:42:41
You can say name of something. Item prop means some items. Property must have a name.
1:42:45
Property key value name. Colon double quotes.
1:42:49
Name. Colon double quotes. Avatar. Double quotes.
1:42:53
Director colon you know double quotes James Cameron.
1:42:56
Key value. Key value. So a movie can be described by describing the name of the movie director, the movie genre of the

movie, and even a trailer link.
1:42:59
So then the movie becomes a rich snippet. So when I then search for the word avatar, we can actually try this.
1:43:09
Okay, if I just say avatar. Then instantly they like, so cool, right?
1:43:14
So all of this came from, you know, somebody describing, you know, this has some kind of a rich snippet being one of those.
1:43:20
It might be an IMDb and I have to go and check. Okay. If all else fails, Google can make up their own.
1:43:27
Okay. Nice. So all this came because they use certain tags.
1:43:31
Now it doesn't make sense. In other words, one way to get all this information is from a so-called actual knowledge graph.
1:43:36
Another way is it doesn't actually have to be a graph graph. It can be those tags that I showed you.
1:43:41
Probably not tags like maybe IMDb does that. And then this can go on pull from IMDb and you don't even have to go to IMDb.com okay.
1:43:46
See? In other words, it's all about saving time for the user in the future.
1:43:54
When when even have a browser like this, or maybe the whole URL would be gone.
1:43:58
Maybe it's all summaries. Okay. And so, um, if you're an IMDb, what do you do?
1:44:02
You do a Google search and you go search. Why not?
1:44:07
Right to an illegal search? Oh my God.
1:44:11
Okay. Exact matches. Hey, wait! The [INAUDIBLE]?
1:44:15
Okay. 92. What the heck?
1:44:19
Okay. Um. So flushed with wrath.
1:44:22
Eldorado. Credits additional crew over the Madagascar. Oh my God.
1:44:26
Okay. Uh, nice. Okay. Okay.
1:44:29
Uh, yeah. What else should I tell you?
1:44:34
Oh, almost to the end. Yes. Movie properties.
1:44:39
So then, if you want to list movies, what tags are available? I'll never ask you on the exam for any of these, okay?
1:44:42
So please don't go in a cheat sheet and copy. I'm never going to ask you.
1:44:47
Name five tags in the movie tag. You know database. It's pretty dumb, but you should know that it's out there.
1:44:51
Yes. So if you use all these tags and give values for them, then your movie would be nicely formatted.
1:44:57

When somebody searches for your movie, other words, somebody came up with all this,
1:45:03
the work of the movie studios and said, what, uh, property should a movie have?
1:45:07
You should have awards the movie you no one should have the author.
1:45:12
It should have audio alternative headline aggregate rating across like many different sites.
1:45:15
Copyright holder. Publisher. If you do all this, then that's the movie, uh, part of schema.org.
1:45:19
So schema.org goes all out and describes all these tags for things like people, places, movies,
1:45:27
events, recipes, books, movie courses, you know, like columns in a table, in a relational table.
1:45:34
Suppose you want to describe all the courses that you see. What columns do you want?
1:45:40
Obviously. Course title. You know who teaches the course? Number of students enrolled.
1:45:44
Average course rating on the right. My professor. You know, stuff like that, right.
1:45:49
So those column names are what we call tags here okay. The tags and column names are the same thing.
1:45:52
They're called properties. They're called descriptors. You know what. What would you use to describe a movie?
1:45:57
How would you describe a course or a professor. And they become keywords.
1:46:01
So in just on your keys and values okay. Okay. Here's what I'm going to tell you.
1:46:05
This very important. All right. Suppose I say a key called a hash value of five.
1:46:09
And then a key called b hash value of USC.
1:46:14
Suppose okay. So then say that's one object, meaning that's one little piece of Json with a comma here.
1:46:17
Okay. Suppose I make another one? Very similar.
1:46:23
But now it's a similar kind of a a big key.
1:46:27
The keys are the same, so it keys a, but the value is going to be different.
1:46:30
Value is going to be two comma. And then the key still called b the value is no UCLA okay.
1:46:34
Okay. And then I have a comma here. And I'm going to have an array here.
1:46:42
And so that's like an opening closing, uh, square brackets.
1:46:46
And maybe I'm just going to call it, like, data or something. Data. And then I'm going to enclose the whole thing in curly brackets.
1:46:50
Okay. Like that. So this is an actual piece of Json file. I can call it like or Json.
1:46:56
But in there there's this key called data. A value would be this array that a hash object object.
1:47:01

It's called Json object Json object. In each object there are the same keys a key a b key.
1:47:06
What this is equivalent to is a table. A table with a column called a and a column called b.
1:47:12
In the first row is value is five, B's value is you are C, the second row s value is two B's value is your salary.
1:47:18
That's what I meant. So this so these things that you see on the left hand side, all of these, all of these,
1:47:28
they're all keys to describe like one object in a one row of data, or equivalently the one column in a table.
1:47:34
Okay. So then imagine how many columns. It's pretty amazing right.
1:47:41
So for things like creative works like, you know, movies and books and forms and artwork,
1:47:44
it looks so many different patterns that people came up with. But it's not some random people from schema.org or some Yahoo engineer or something.
1:47:48
It is actually creative people. They go to a bunch of artists and you say, how do you want to describe your artwork?
1:47:56
They'll say things like dimension, medium, like oil, acrylic in order to rank, um, theme.
1:48:01
You know, what kind of a subject is it? Is it still life? Is it landscape?
1:48:07
Is it, you know, portrait? So artists will tell you these things are important to us.
1:48:11
And so then you make tags out of them and then you can start describing artwork.
1:48:16
Okay. So go ask the people who are domain experts. They're going to tell you all right you guys.
1:48:19
So that's all. Otherwise there's not much more to say. There's like a mind numbing amount of content.
1:48:23
So the classic the prime example then is I want to describe Pirates of the Caribbean.
1:48:27
This uses by the way, is uh, XML. XML is all these sharp brackets where you have to say them and you have to say slash div.
1:48:33
You need to, you know, say span. You need to say slash spent.
1:48:41
XML is an older format. Json is a much newer format.
1:48:46
All of external can be 100% replaced by Json. Unfortunately, microdata was born before Json came along, so it is stuck.
1:48:50
Connection maligned okay, but eventually convert XML to Json and vice versa in XML when I have a, uh, key.
1:48:57
So this is item prop author, right? And then it says item type person and then name is Ted Elliot.
1:49:06
Okay. So what I'm going to do is, uh, author okay.
1:49:13
I'm trying to see how I can easily represent.
1:49:18
I'll just do this little part, you know, rather than all of this in Json, it will be simply double quotes name colon and then the value shall easy.
1:49:20

That is okay. Likewise it'll be double quotes rating value colon eight.
1:49:31
Throw all the other crap away. XML is very chatty.
1:49:38
Too talkative. You know there's way too much of the type in Json is very minimal.
1:49:41
Okay, okay, but otherwise one can go and become the other.
1:49:47
In the end. You're saying that every movie needs to at least have these as a minimum.
1:49:50
You can have a lot more about the movie, okay? Like how much money you make and how much it cost to make.
1:49:55
But as a minimum, everybody in the world wants to know for a movie what is the official name of the movie?
1:49:59
And some simple description, what the movie is about, who directed the movie?
1:50:04
That's important. A lot of people who wrote the screenplay, you know, if that's the only thing.
1:50:07
And who are the actors and actresses and what is the average rating the movie got?
1:50:11
That should be enough to decide if you should watch the movie or not. Right? Okay.
1:50:14
That's all. So this is what I call column names okay.
1:50:17
And these become enshrined in actual microdata format, which is mind numbingly, you know, painful, complex.
1:50:20
And in the end, you will have a table somewhere and somebody would fill in the blanks and hit enter,
1:50:26
add data, and then that all will be generated and become part of, you know, a page that is microdata.
1:50:32
Then when the search engine comes and grabs all this, searching through the microdata and pull out all the things like, you know, author,
1:50:38
director and then format them properly for you so that when you go and search for things, then you will actually start seeing them.
1:50:44
That's actually what happened in avatar. Do you see that? I said the word avatar actually back here.
1:50:51
See this, all this. So it came from like budget, sequel, box office,
1:50:56
released a director and all the cast is because all those were described using micro data and Google cannot make use of it.
1:51:00
In the end, it's all about snippets like how like what goes into generating snippets.
1:51:07
And the answer is structured data goes into generating snippets, then no errors involved.
1:51:11
Okay, some more examples, you know and it's just a little bit more detail right.
1:51:16
Things like time okay. So time can be described in so many different ways.
1:51:19
So you just got to pick one. These are actual calendar times right.
1:51:23

But look at this one one hour 30 minutes. It takes time to cook.
1:51:27
So you need to say T1H 30 Am. You know so actually is some kind of link time 180 is one hour 30 30 minutes.
1:51:31
No gap between them okay. So you have to obey the format and then it knows the cooking time is 90 minutes okay.
1:51:40
So in chat jeopardy you can say tell me all the recipes that take 90 minutes or less to cook.
1:51:45
It should hopefully find this one. Great. And that's all.
1:51:49
So our concept can be likewise described by uh. You can put it on, you know, Taylor Swift if you want.
1:51:53
Right. Us. All right. So date time in our Markov enumeration.
1:51:58
Enumeration means, you know, like an Amazon, the trying to solve something or based again column names.
1:52:04
What is the name of the item? You know, what is the manufacturer. What is the one line description.
1:52:09
What is the price? How many units are left. You know yeah. What is the Amazon rating.
1:52:14
All that can be called enumeration. All right. They're all item props.
1:52:19
Again name column that price column that availability column that end of story Json is always simpler.
1:52:22
Okay. So then in here again, you know, you can actually generate this yourself.
1:52:32
Meaning there's a form, you know, that you can go and look for.
1:52:35
It's an merkle schema generator where you would use a form, literally a UI to fill all that up.
1:52:38
Cells, you know, put all this right. And then you can even put in like social profiles you can pick.
1:52:44
So all of this will be automatically converted to that. So you don't have to type all these keys and values by hand.
1:52:48
In other words as a front end as a form okay. And then you can then grab that, you can copy and paste it into a web page.
1:52:54
And then suddenly there's your profile and then the search engine can come and grab it.
1:53:00
Okay. In other words, we don't need to manually type all this.
1:53:03
You can have a UI and the URL spit this as an output,
1:53:06
but that in turn will go into a page that a search engine in the future will come in index and grab all the data from it.
1:53:10
That's all because if you make humans type this, ultimately what that sign by mistake.
1:53:17
Or they might leave out the right double quotes, or they might forget a comma if you do any of those is not valid Json anymore.
1:53:21
Search engine will basically ignore it. Okay, it's not going to try to work around your syntax error seven and nine.
1:53:27
We'll do something in just a few minutes okay. All right. So again you know rich results.

1:53:32
So this just means you know do you actually have rich snippets in your page or not.
1:53:38
You can actually go to a page like that and put in your URL. It will tell you how much rich snippets you can extract from your site.
1:53:42
So you can then test your own page to see if your page is rich, snippet friendly or not.
1:53:48
Okay then if they say that page is eligible then you can start putting all this rich nepotism can try on.
1:53:54
Okay. I want to finish this and do something.
1:53:59
Okay. So again, the whole testing tool, you know, you can put in any URL you want and say test my structured data.
1:54:03
Then it basically parses your page and pulls out what could be, uh, rich snippets and showed it to you.
1:54:10
Like this. Very interesting. Right from your page. It found that there was a restaurant, you know, and so on.
1:54:15
Right. And then they call it a testing tool. Okay. That's all.
1:54:20
And it tells you that in here these are very useful. They're useful for business owners okay.
1:54:23
Because the last thing you want is errors in passing because somebody put some, some bad data formatting.
1:54:28
And then the search engine bypasses your pizza restaurant and never shows up in anybody's search.
1:54:34
You lose all the money. And so where my customers know if you speak English, where are my customers at?
1:54:38
Okay. Don't say that. Okay. All right. So that's all.
1:54:45
Okay, so what else is testing? This is more of a testing tool, you know.
1:54:50
I'm skipping all this. It basically warns you, you know, about missing data or something like that, right?
1:54:55
Snippet testing. Okay, so new tags for snippet control.
1:55:01
Uh, okay. So, you know, supposing. Okay, what if you don't want your page to be a snippet?
1:55:05
It's the opposite of what we started with. Many people want the page to be part of a snippet.
1:55:11
Right. But then suppose you tell Google you want to tell search engine.
1:55:15
Leave my page alone. I don't want to contribute their snippet like robots, right?
1:55:18
Previously we had robots to text. You could tell robot the crawler.
1:55:22
Leave my site alone. Don't crawl. Likewise, when you say no snippet in a meta tag or maximum snippets, and you put in a number like four,
1:55:26
it means you're telling the search engine you can only make up to four snippets from all this data.
1:55:34
Not more than four. Not more than two. Not more than one. Not more than zero. Likewise.
1:55:38

Video preview. Okay, that's very interesting. So say your search has video.
1:55:42
And how much can they put in the preview. Can they put the whole video in the preview. Probably won't like that.
1:55:46
So you can put in how many seconds. So the preview has to be limited to that many or fewer seconds.
1:55:50
So all these ways of you having some control likewise image preview you know.
1:55:56
So what about the size of the image. Can it be small. Can it be large.
1:56:00
Can be one. None. So they give some control back to the content creator and says the whole snippet extraction is not automatic.
1:56:03
Very cool. Right. And then it's also a robot's specification.
1:56:10
So you say the word meta and then this exactly how all that is specified.
1:56:14
Max snippet 50 image preview large. Then it will do what you want.
1:56:17
Okay. You can say no if all of them. And then there is no separate snippets.
1:56:22
Okay? You told it to go away. All right. So that is all like question again.
1:56:25
You know, just more and more. Uh, this is actually very interesting.
1:56:30
Span can be put anywhere, right? You guys know about spans okay.
1:56:33
I'm just going to do one span for fun. And also this is actually very neat.
1:56:37
So you know I'm not going to go there yet okay. That's what 585. But check this out.
1:56:41
I'm going to go to bear. This is what I mean by a span you guys.
1:56:46
Check this out. This is actually a pair actually do a paragraph tag actually.
1:56:50
So look, if I do a paragraph tag and go all the okay.
1:56:55
So the paragraph. Right. But then just for any little part of it I can make a little span.
1:56:59
I can say the word span and I can say the word span. So then span is a mechanism that you can insert in any tag inside the middle.
1:57:05
Open and close it. So in the span you can say no snippet. So then for some reason all would not be part of any snippet.
1:57:13
That's basically what the range is okay. But span as a much better reason to exist which is a style specification.
1:57:19
You can say style equal to. You can say font size, font size.
1:57:25
Um, something like, you know, 100 pixels something. Right? 100 semicolon.
1:57:30
Now suddenly what happened? Wow. So just that little span you know, of, all right, that the little while span.
1:57:35
I gave it a style of 100, and then it became a hundred pixels to Spanish.

1:57:43
Very neat thing. It means you can go in a piece of a paragraph or a table in,
1:57:47
or a bullet item anywhere and make a little span and have the span have its own style properties.
1:57:51
In this case, font size. I can do font color here, and then I'll write all that, but that's what the idea is.
1:57:56
But here they're abusing, so to speak, the Span specification to say span data.
1:58:01
No snippet. It means you're telling search engine whatever is in that span.
1:58:06
Don't make it part of any snippet code. Right. That's also, you know, there's no snippet tag, all snippets data, no snippet.
1:58:10
That means only in feature snippets. When you say no snippet, it means it won't even be a regular text snippet.
1:58:18
Okay. When you say data snippet,
1:58:22
then it might be in regular snippet the text but it won't be in featured snippet won't be like highlighted showcased specifically.
1:58:25
So like all kinds of, you know, I guess things to go along with that.
1:58:32
All right. And then here's the summary. You have regular snippets which is a URL.
1:58:36
And then some sentences, some words that follow that the URL rich snippets, they come from things like recipes like all that.
1:58:41
Right. And then media event and then new snippets and then entities.
1:58:47
Because I know about entities from the knowledge graph usually and then features, you know, then they put that in a separate box promptly.
1:58:52
And even people also ask. I think that's also part of the feature snippets.
1:58:58
So even though you didn't know this all these years, it definitely consumed it. But now you know it has a name.
1:59:02
What you are looking at is called a snippet. And then search summarization. Bell alarm will be more about snippets.
1:59:06
What Google and Microsoft are all betting on is that this format of showing you raw results, right for pages on end, is a thing of the past.
1:59:12
Most people will be happy by just reading the little summary because they're so good and go away.
1:59:21
But I think it's a pipe dream. You know, in fact, I think it will backfire against them because when more people use it,
1:59:26
they'll discover, wow, it lied to me or gave me the wrong prescription, okay?
1:59:32
It killed my baby, whatever the [INAUDIBLE], right? Then people will, like, push back and go, oh, my God, just give me back my search results, okay?
1:59:35
So the best is obviously to have both of them. It'll give you a summary.
1:59:41
And then at the bottom it will tell you all the links from which the summary came from.

1:59:45
So something in the summary says then you can go and click on the links yourself and look at it okay.
1:59:48
When you write an academic paper you can claim whatever you want in the paper. And then you have references because you're telling the world.
1:59:53
Yeah, I got all my stuff, but I got a lot. From here. Same idea. Okay.
1:59:59
So I think hopefully there will be always a need to go back to all links. Okay.
2:00:02
But then actually the companies are betting against everything people want like Rawlings anymore.
2:00:06
We'll see. So then then the whole paper is also a kind of, uh, snippet, but then they can go in the schema.
2:00:10
Or you can do this yourself. Okay. In schema.org they have more, um, tags and in the tags,
2:00:19
if a site like Reddit or StackOverflow takes all the topics that they think are very similar based on the similarity,
2:00:26
and then they use those tags in schema.org, then those will become the little box.
2:00:32
That is, people also ask. So this answers the question how does the search engine know what to list.
2:00:38
And people also ask. The answer is useless tags.
2:00:43
And then for the value. So for these keys use those tags for the values.
2:00:46
Use actual core account question or StackOverflow question somebody asked.
2:00:51
Okay great. And how do you know what to put in. That's a recommendation engine okay.
2:00:56
Meaning that's a similarity search. Cool. Okay. Then Monster.com has an article about it.
2:01:00
Wow wow wow. Okay, so not bad on time. A little bit behind, but you know.
2:01:07
Check it out. And now we do this and then go.
2:01:11
Oh, we do the Macarena. Oh, Macarena. Okay, so why do we do the Macarena?
2:01:17
Because. Uh oh. Oh, no.
2:01:22
HB sharp. It's racial.
2:01:27
Either completely absent or send me a massive pile of emails and hyper focus.
2:01:32
I'm going to be out today. Or is here only three possibilities?
2:01:37
Okay. It should be shot. Okay. So if you are actually violently absent, please mail me when you watch this video.
2:01:42
Otherwise. I have a list for five points off. Don't worry.
2:01:50
Okay. Wonderful. Gangs.
2:01:55
El. Gangs out. I don't remember getting an email from gangs and I might be wrong, but I don't think so.

2:01:59
Oh, you can just randomly be absent. Okay. There's only five more weeks, including this week, so please come to class.
2:02:06
I come to class. I could be at home doing all this remotely, right?
2:02:12
Come on. Okay. Okay. Uh. Gee hug.
2:02:17
Yeah. Is said. Gaga. GA GA hug. God, I'm so sorry.
2:02:21
Okay. It's a pretty racist sounding, very funny thing.
2:02:25
So I won't play it obviously here, but in there to make fun of names.
2:02:32
Okay. Exactly like that. Okay. They would not call somebody God.
2:02:36
God, they would call it hog, you know. So but I'm not going to do it by the way.
2:02:39
Again I'm not going to play it. But I'll tell you, please go on Google a YouTube video called The World's First Racist Self-Driving car.
2:02:45
Okay, that's obviously a comedy that was made on purpose to push down racial stereotypes a little bit, but in their own minds,
2:02:55
as you mean, they actually made a car that specifically kills black people because it actually what the car does.
2:03:01
Okay, in the beginning of the clip, a guy, a black guy, is walking on the street.
2:03:07
The self-driving car keeps on going and hits the guy because it cannot detect a black person.
2:03:11
It's w t f okay. I mean, that's basically the point they're trying to make, but it's called the world's first racist.
2:03:17
So it's funny, but in a sad way. Okay. But yeah, stereotypes can be funny because they're painfully true.
2:03:22
You know, they keep happening over and over and yeah, like, you know, wow, I might as well laugh about it.
2:03:28
I make joke about it. Okay. So, um, at Huggy r the G.
2:03:33
I'm so sorry. Okay. Who's that? Okay.
2:03:39
That's cool. It always feels like somebody is raising their hand. You don't look at it.
2:03:45
Look behind. Okay. See that? Let's go to like here.
2:03:48
Okay. Or armies of the world tonight.
2:03:52
People bar one of those. Teach ten.
2:03:56
Should be an equal. Thank you. One more. Bishnoi it.
2:04:01
Bishnoi. Missionary. Visionary.
2:04:06
Visionary. In some states, B and V are interchangeable.
2:04:10

Okay. Like West Bengal.
2:04:14
Vengo, Bengal maybe. Who knew where Bishnoi originally?
2:04:18
Going once. Going twice. Gone. Cho ruddy.
2:04:24
That's it. Wow. Okay. Why is it not so dark?
2:04:30
That seems like there's a little bit of, uh, dyslexia going on, right?
2:04:33
Which other? Right. Okay. Chocolaty.
2:04:37
I'm not trying to make fun of your name, but I know a little bit about names, so, like, I guess this name would be a lot more common.
2:04:41
Hey, this is a great segue into our. The whole Levenshtein algorithm is do a Levenstein distance.
2:04:47
Edit distance between Chaudhary and Chara. See how this Chaudhary.
2:04:54
There are many others. Okay. Hey, wait. We're going to do this. Okay. Chaudhary Chaudhary is 200,000.
2:04:58
So, Ravi. What is your guess?
2:05:05
For photos and not for later photos. Wow.
2:05:11
Okay, that's a little bit less common, would you say? Check that out.
2:05:15
Yeah. My name would hardly come up. Okay, so I shouldn't be talking.
2:05:19
All right, so we should do the next one. It's never too often, right?
2:05:23
We should know the query processing care so much about snippets.
2:05:27
By Joseph. I'll do any song over again.
2:05:31
Okay, so then we should do our next topic, right? Query processing.
2:05:37
Only 32 slides, not 44. This should be an easy topic as well in a way, because we're just saying optimizations to the the engine underneath.
2:05:41
You know, there's already definitely an indexing involved indexing algorithm going on constantly.
2:05:49
You also know this page ranking going on okay. So is that all there is?
2:05:55
In other words,
2:05:59
would the system take your query term going to inverted index and come back with a bunch of pages and pages already have page rank computed for them?
2:06:00
By the way, you don't complete it after you search and then rank them by the page rank.
2:06:08
That's why it's called page Rank and show you the top most rank page first.
2:06:12

Then second, third, fourth. Is that all that is happening? And the answer is absolutely not.
2:06:16
What else is happening? That is what the slides are for. And it's very fascinating.
2:06:20
It's simply a history of Google's evolution when page rank was first invented.
2:06:23
That is actually all there was, which is there was a web crawler that crawled a whole bunch of pages,
2:06:29
indexed it like your homework, and then those indexed pages got ranked.
2:06:35
You got page rank, the page rank, restored, the page URL somewhere.
2:06:39
Then when you search your searches, match to it, you know all of the the inverted index URLs and then it'll pull up like a bunch of them.
2:06:42
Right. And then sort them by page rank and literally display top to bottom, highest ranked low rank.
2:06:49
But the biggest change that they made was they have this new things called signals authority signals.
2:06:55
In other words, not all sites are equally reputable.
2:07:01
So there's a very shady site that's a very good site.
2:07:06
So and the bottom oh wow. That's literally chain number.
2:07:10
I'm not going to take it okay. Okay. So in a class or I'm going to go up.
2:07:14
So you know I'm just going to say it when you have two different sites that have the exact same content,
2:07:18
then regardless of page rank, you should not treat them equally because a bad site is literally bad.
2:07:24
Okay. Even though the links might make it seem like they have a higher page rank.
2:07:30
So we need to punish what are bad. It's like blacklisting, you know, certain servers.
2:07:33
Okay. And then anything that comes from that is spam automatically will put you in the spam folder,
2:07:37
because past experience tells us that this is a bad spam site.
2:07:42
So likewise, you can actually blacklist certain sites and make their so-called goodness factor a goodness.
2:07:45
How good is that site G a value called g? That could be very low.
2:07:51
That's empirical. Google actually collects data, you know, over time, and then you add that g value to things like the page rank value.
2:07:55
So if a site has very high page rank and a very low g value, you add them together.
2:08:02
It might go lower than some other page that might be much, much better.
2:08:07
Good is high and page rank is not that great. But so good is so high that it will actually win.
2:08:11
Okay, so the good guys win. Basically, that's all this is. So what are the so-called signals okay okay.

2:08:16
So yeah you can do more things okay. So these things are going to do list for you.
2:08:21
One is if you go in the textbook, you know, we actually have a textbook, right.
2:08:27
As you know Geoffrey Allman's textbook in there. You can do data structure reorganization just to make up the inverse index lookup faster.
2:08:30
You can reorganize some of the data. That is a mechanical thing we can do.
2:08:39
That is the first thing that any search engine would do because you get the gain for free automatically.
2:08:42
Why would you not do that? Okay. Uh, so we'll look at a little bit about that.
2:08:47
And then also this is ultimately what we're trying to do from the outside.
2:08:51
We are trying to guess what Google does.
2:08:55
And every year the constantly modifying like how search works probably every day, even though every testing as we speak,
2:08:57
there's a small number of experimental users of a new version of the engine that we don't have.
2:09:04
And if people seem to like that, they like be over a little switch all of us to be.
2:09:08
If people don't like it, go back to it.
2:09:13
Okay, so this thing is not as, um, a static target that the whole world can go and hack the constantly modifying what they do.
2:09:14
So best we can do is reverse engineer from the outside. We can guess what Google is standard.
2:09:21
Okay. You can look at that. And then yeah, this is also to the extent that they publish like how this stuff works.
2:09:25
This is the last few slides okay. In here. See this Rankbrain okay.
2:09:31
So PageRank is replaced by something called Rankbrain. So Rankbrain has a patent as well.
2:09:35
So that's what Roland lets you call. So again three parts to like what we're trying to look at.
2:09:40
And they all go very fast. Yeah. So and then write the user type something I go and type like all kinds of things to avatar.
2:09:46
And then you want to return the terms. Right. And then you also try to guess the user's model.
2:09:54
I could be learning Sanskrit, and I want to know that the word avatar means incarnation, but that is not the.
2:09:59
That's not the meaning that most people know or care about. Don't know about the movie, right?
2:10:04
But if they know from a past search, they search for lots of Sanskrit words.
2:10:08
Then maybe when I type avatar it might give me the Sanskrit words meaning okay, that is what is meant by trying to guess what the user wants, right?
2:10:12

That's very difficult. Also, we don't usually, you know, state to the search engine exactly.
2:10:20
What we want were pretty vague. Okay. So they have to basically read between the lines. There's a lot going on.
2:10:25
So they can use a knowledge graph and even your Wi-Fi location and your profile that you put in.
2:10:29
You know, I'm a student, I'm a professor, I live in Los Angeles. You know all that, right?
2:10:35
And so they can use all of those. So these all become what are called signals.
2:10:38
Okay. So it is not just simply PageRank. It's way more than page rank.
2:10:42
I want to tell you. Great. So what are all the heuristics and how can speed up okay okay.
2:10:45
Strategy one. Yeah. What about this one. Only query terms with high IDF scores okay.
2:10:50
So not all query terms how these are you know you remember IDF right.
2:10:55
Inverse inner documents sequentially you have term frequency also in the document frequency
2:11:00
in how many documents of all the documents are term appears near to one over that.
2:11:04
Right. So this is a very interesting, uh, example.
2:11:09
Catcher in the Rye, by the way, please put up your hand if you read The Catcher in the Rye.
2:11:13
Anybody? Sorry. I'm nobody. Writer. Okay, I read it, obviously.
2:11:19
Please, please, please, please read catcher in the Rye.
2:11:23
It's a beautiful book. It will make you cry. Okay. But it's an amazingly powerful.
2:11:27
It's a great, great American classic. Okay. Should read it.
2:11:31
It's also a so-called banned book. There are so many libraries in the US that don't carry the book.
2:11:35
There's nothing subversive, not even one swear word. There's nothing horrible, okay?
2:11:40
Nobody gets killed. But still the ban in shocking. All right.
2:11:43
So then in the search for the book, then you should only look for not stopwords, but words located right?
2:11:47
Right. Are basically they are the words that make more sense. Correct. So because those other words are in like basically every single document.
2:11:54
So we shouldn't try and do some combined PageRank or something with all of them.
2:12:00
So only rank meaningful words. That is all. Okay. Yeah.
2:12:04
So low IDF terms, you know, because if the term occurs too high, if some term like in occurs in every page, it's a pretty high number, right.
2:12:07
One over that is a very small number. So then eliminate those words.

2:12:17
That's another easy an easy way to say that is eliminate stopwords.
2:12:21
Don't even look for them. In The Catcher in the Rye example, the only search for documents that have the word catcher and right,
2:12:25
and probably not too many documents have that except the book reviews.
2:12:31
Maybe you and the PDF of the book. By the way, you can do very crazy cool things like this, right?
2:12:34
You probably know this. You can type catcher in the Rye, catcher in the Rye.
2:12:38
As soon as I say the catcher, that's really all there is, right?
2:12:43
You can type words like dot, PDF, okay next to it and then suddenly somebody because notebook.
2:12:46
See there it is all of 128 pages okay. It's a great great great book.
2:12:52
I'm not going to read it looks like that okay. Yeah we read it this weekend.
2:12:55
But when you say the word uh, PDF then suddenly it starts showing all these cool PDF links, right?
2:13:01
Hackensack Public School, school, new Jersey. Nice. Okay.
2:13:07
Now or this only an excerpt. Wow. Only 46 pages.
2:13:11
Yeah, exactly. Okay, so then, uh, we can do that, and, uh, so avoid stopwords.
2:13:16
Okay. That's easy. What else can we do? Yeah.
2:13:22
So, you know, what about docs containing several query?
2:13:26
I mean, you have multiple terms, right? Then you should, um, look for, you know, cosine score scores for as many terms as possible.
2:13:29
There's a reason why the user typed all this terms, right? So then consider documents in which basically as many terms as possible occur.
2:13:39
That is very cool. Several times. That's pretty obvious, right?
2:13:46
Because otherwise I wouldn't search for all those words like say I type something like HashMap, Java data structure or something.
2:13:49
Then I don't want a HashMap in some other language like, you know, Ruby or something, right?
2:13:56
I want Java and I say I want not hash map source code or something.
2:13:59
I want to know like what kind of data structure is it? So please take everything that I type.
2:14:03
As long as it's not a stop word look for like, you know, similarities for all those words.
2:14:07
That's what this says. So these strategies are very obvious in a way.
2:14:11

Right. What does the word conjunction mean? What does the word disjunction mean.
2:14:14
What is the word conjunction mean. Conjunction means.
2:14:19
And exactly this junction is all right. So in a way, secretly, it's almost like you're saying Java HashMap data structure.
2:14:25
I want all pages that have the word Java and HashMap and data structure, even though I don't have to say end and it's automatic.
2:14:32
Okay. And then postings traversals. You know what postings mean for every word.
2:14:39
A list of all the URLs that the word is in, it's called the postings list.
2:14:44
So then we can then combine all the postings list. Right. You know, otherwise get the postings lists where uh,
2:14:47
the documents in the postings list will have basically all the search terms, if not most of the search terms call.
2:14:53
All right. And then this notion of a champions list, you know, okay, this is also cool for any term at all for any term, including our data structure.
2:15:00
Uh precompute. Google has so much infinite computing power disk space to store all this, right?
2:15:10
They can do it. They'll already pre-compute for for any commonly used word like data structure.
2:15:15
What are the good documents to serve for anybody? Searches for data structure.
2:15:20
We call it a champion list. Okay.
2:15:24
So our champion list is not every single page that has a data structure that leaves some other pages out because are useless.
2:15:26
Really? So then that makes the query easier to serve because when you then type data structure.
2:15:32
Okay. The 1 to 1 go and look through the whole big index that they have.
2:15:38
They look through the champion list and see if they made an entry for the word data structure.
2:15:41
If there is, return all the URLs from the champion list for 1,000,000,001.5 billion.
2:15:46
If the, uh, phrase or the word is not in the champion list, then they can go in the row index and look for it.
2:15:51
Okay. And if enough people search for that, then they'll consider making a champion list for this.
2:15:57
And how would then I go searching for what? Man, they have the law for everything in the world.
2:16:02
Okay. For every second when people are searching, Google has logs about what people are looking for,
2:16:06
and they analyze the log constantly and pull out trending keywords and on top of terms, okay.
2:16:11
All right. So that is an easy. These are all obvious strategies right.
2:16:16
Just spend some time on them. And also this is very interesting right.
2:16:20

This notion of are so all documents of the highest.
2:16:24
We are just some number like five you know 100 uh for some term.
2:16:28
But sometimes it might actually happen that the Fraser searching is so rare that
2:16:33
if Google decides the number called k for every search will return 100 documents.
2:16:39
Okay, so case 100, it might be that the R for your search term because your system is so specific.
2:16:43
Uh, macromolecular lipids or something. Right.
2:16:50
Then that way the r the good number of in our champion documents would be smaller than even like what, the one that potentially served you.
2:16:54
So I can be less than k. Okay. It's a k, by the way.
2:17:02
Like I said is 100 or something where they predetermine how many docs to return so I can be smaller than K.
2:17:04
And that does not mean anything is not good is not bad. It just means that what you search for is pretty rare.
2:17:09
Okay. Yeah, yeah. Then that's it. So then I'll only consider the champion list if you have a champion list.
2:17:14
So all these were said over and over and in the champion list.
2:17:20
If you have larger documents more than K then pick k. Otherwise less than K.
2:17:24
Pick picture. That's all. Okay. Know then if statement if less than k.
2:17:29
So all of it. If r is less than k. If r is more than k then picture.
2:17:33
Okay cool. I don't know. Someone's keeping on talking in the.
2:17:39
Okay, so please don't talk to guys. I don't know is talking. I'm.
2:17:43
Doc, please. The people that are sitting next to you, by the way, they're highly bothered by you.
2:17:46
And close your eyes. You look, I'm closing last week in the whole world.
2:17:52
I cannot without using my hand closed my ears. Can I do it?
2:17:56
Okay, so people, next you are trapped. Basically, you have to listen to your talk.
2:18:00
Unless you can do this all the time. But then they cannot hear me talk.
2:18:04
So please don't talk. All right, so quantities cause this where the whole goodness, all that comes in.
2:18:07
Right. Okay. So on the one hand you have relevance okay.
2:18:13
Which is basically Tf-Idf, right. Based on just the words themselves.
2:18:17
But beyond relevance, what about things like authority.
2:18:21

In other words, the same term exists in two different pages.
2:18:24
One of the misfortune, you know, our daily call, our daily signal, or some radical right on the left and right.
2:18:28
The other one is National Geographic, Smithsonian, Washington Post, New York Times.
2:18:34
Okay, BBC. Then they'll naturally pick the things that are named towards the end.
2:18:39
Like Smithsonian, you know, uh, Washington, you know, National Geographic.
2:18:44
You might argue it that you might be a right wing person. Oh my God.
2:18:48
And know this whole brainwashed, you know, this mass media. Okay, Jewish liberal media, that's what the Republicans sometimes call it.
2:18:51
Well, you're not too bad. Deal with it. Okay. So in other words, Google basically has to make a choice.
2:18:58
You know, they cannot please everybody.
2:19:02
Uh, so they say sites like National Geographic, you know, they've been around for like a long, long time before the government came along.
2:19:04
So we trust them. You know, they're not going to screw up people's lives. Right.
2:19:09
So therefore they'll have higher authority. And that authority is a score.
2:19:13
That score will be added before you rank everything. So the higher the authority those pages will go up and ranking you will see them first.
2:19:17
Okay. You can literally see this as we speak. Suppose I say the Great Barrier Reef or something, right?
2:19:23
So look, I just have the Great Barrier Reef just type someplace, then instantly it'll give me about right.
2:19:29
Unesco World Heritage, UNESCO's a worldwide organization that once protected by the way, because of global warming, it is getting bleached.
2:19:36
All the coral, they're dying. They're getting like white, bleak, dead under the water.
2:19:44
It's pretty sad. And then obviously there's an organization to pick it up on. Wikipedia.
2:19:48
So highly trusted site. Right. Australian tourism. They actually want you to go there.
2:19:52
Encyclopedia Britannica. You see, these are all trusted sites. Okay.
2:19:56
There's a reason why you see them at the top. The authority is very high.
2:19:59
And so that is basically what this slide is about okay. So if I make a thing about red berries I might be cool.
2:20:03
But they might put me somewhere at the bottom. You saw right. National geographic, you know, again highly reputed to register more than 100 years ago.
2:20:09
They're not there to lie to you. Right. So Wikipedia, newspapers, you know, also page ranking.
2:20:15

Okay. So if you make a page, you know, that is cited by a lot of other people, it means that's the classic page rank.
2:20:21
Okay. Yeah. So these are the same. It means you you have such good content that others tend to refer to you.
2:20:27
And so then you have to go up. So there is an authority. In other words this is a new authority.
2:20:33
New authority. This is classic PageRank based authority. So 30 value can come from so many different sources.
2:20:37
That is all. All right. So um yeah.
2:20:43
Then how do you do that already. Measure some kind of a quality score okay.
2:20:47
Which is 0 to 1. Imagine you have something called a quality score.
2:20:51
You call it G. G stands for goodness. So goodness. Favorite document.
2:20:54
So depending on the origin of the document, the recommendation National Geographic, the document is in some other, you know, questionable side.
2:20:58
Then the goodness value for the document. This value for every single document would be either high or low but never less than zero.
2:21:05
Never more than one. Okay, so good documents would have a high g value.
2:21:12
You know, shady sites. Okay. Such sites are going to have, you know,
2:21:16
pretty low g value and there's nothing you can do about it if you don't like it, go start your own search engine.
2:21:20
Okay. Yeah. Not happening. All right. So then again in a how do you do what.
2:21:25
And this over and over okay. Yeah. Good. So now you understand.
2:21:30
And that's what you do. So this is classic standard document similarity coming from IDF okay.
2:21:35
But now additionally you add a goodness measure and that can make rankings go up or down.
2:21:40
If two pages have the exact same cosine ranking, you know, say there's some kind of a query term here.
2:21:46
How do you how do you do it? Okay. So if you have a query term here and there is one document here, second document here,
2:21:53
they both have the exact same Euclidean distance, some same cost angle.
2:22:02
Right. That data is same as that data from analogy.
2:22:07
And they're both basically equivalently the same document really when it comes to just that measure.
2:22:09
But one of them has a G value that is pretty high 0.7.
2:22:14
The other one is a g value pretty low 0.01.
2:22:18
So certainly this one be sure that will be sure or they both want to be.
2:22:22

So uh, this will go above this high low. You have to rank it.
2:22:26
So that's all. So net score is just simply G plus course sentiment cannot this cannot be simpler than that.
2:22:30
Right. Okay. Cool. Um yeah.
2:22:36
You know if you want they can play with it in the future. Right. They can say, you know, cosine similarity is more important.
2:22:40
This means half of this plus half of this right, is an implicit 0.5, 0.5 equal weight factor in the feature.
2:22:45
They can make it 0.3 plus 0.7 or the other way around.
2:22:52
They can say, you know, come on, this whole cosine similarity is getting old.
2:22:57
Will make this point three weightage. Point seven weightage.
2:23:00
Let's call what is that called. You average different numbers or different weights?
2:23:04
It's called. One. Yeah, it's called a weighted average.
2:23:09
Exactly. So in this case, the weighted average is equal like divided by two, you know.
2:23:13
So you want the whole thing that one. Then what I'm saying is you multiply this by half multiplied by half.
2:23:18
But now you don't have to add half and half. You can make like a nonlinear non non-uniform like weighted average.
2:23:23
Cool. That's really all. So then you take that net score and then take the top.
2:23:29
How many documents that match the net score. And you say it together.
2:23:33
In other words it's a trivial but extremely important modification because this modification is somewhat subjective, right?
2:23:37
It is not in the document you're talking over the source or the document.
2:23:43
Then host it aside. National geographic is better than for China.
2:23:47
Well, because the world. Okay. Yeah. Look at when they started and what they do.
2:23:51
And so that shouldn't say you you cannot claim the equivalent. Okay.
2:23:56
Yeah. Easy. All right. So that is it. Um, so.
2:23:59
Yeah. Recognition. You know, this is actually very interesting, right?
2:24:05
What if you actually take all of your, uh, inverted index that you already have and actually sort, you know,
2:24:09
order the postings by the goodness measure so that at the very top of an index,
2:24:16
only the good documents are there and all the so-called bad documents are gone to the bottom.
2:24:21
And then that way they can pull what they want to give you from the good list, so to speak.
2:24:26

When they run out of good stuff, then they start going to the bad stuff. So use goes a way to resort, you know, the the index.
2:24:30
Okay. The index can be alphabetical, right? Obviously this can still be alphabetical, but maybe within each alphabet,
2:24:37
maybe even within each search, you can take all the documents that match that search term,
2:24:42
like the word data structure and have all the goodness, high value good documents, first,
2:24:47
bad documents at the bottom, and then only serve the good documents until they run out.
2:24:51
Okay. That means if you go on scrolling, you go to page ten, page 20, pages 32, and then afterwards they have to start giving it a bad document.
2:24:55
But at that point, you know, you wanted it. All right. So that's all.
2:25:01
And then this one is about okay. So the highest J value should be at the top.
2:25:05
That's it. Okay. And then you can do all this as usual.
2:25:09
The same cosine squared term. Nothing changed really. And then computing that score one more time again over and over.
2:25:13
You see like this right there is this. This is the same formula as this one here we said g plus cosine.
2:25:19
And now this expands what the word cosine actually means here.
2:25:25
This is a dot product right. Again a dot b um divided by length of a times length of b.
2:25:29
If a and b are normalized then you would actually simply that's already like length one length one.
2:25:35
Then we don't need this one. By the way, in the paper that I'm going to show, you have questions.
2:25:40
The whole value of question similarity D one question, stuff like that.
2:25:44
You know, I think that we should not be dividing by the length. We should not try to normalize them.
2:25:48
They argue that some query should be left as it is unnormalized okay.
2:25:52
So normalizing actually causes the problem. And all kinds of things are going to show you.
2:25:56
Pretty neat. Maybe tonight. Okay. So then this the normalized value.
2:26:00
Otherwise I've said all this to you before. There's no need to tell you anything more.
2:26:05
There's nothing new here. Over and over. Same thing. Really. Okay, so the notion of.
2:26:08
Hi. Yeah, this. That's actually very interesting.
2:26:12
What if you make artificially two lists called the high list and the low list, basically for every single search term.
2:26:14
Right. And so high is the same as a champion list. When I search for something, I have a good list of documents to give you.

2:26:20
I'm calling that the high list. And then I also have a so-called bar list.
2:26:26
You know what? The questionable results. I'll first give you all the highlight documents.
2:26:30
If you keep asking for more than, I'm going to start getting a lot less documents.
2:26:34
So we call that the high list and the low list. Okay, that is all.
2:26:37
So then when you look for queries postings, meaning when you search for a query and all the URLs only do high list first,
2:26:40
if you already have enough of what you need to return to, the user can take that care like hundred to 100 and stop.
2:26:47
But when you run out, you know, I suppose in the high list is only 20 documents, then you have to go to the long list.
2:26:53
Also, if k is 100, meaning I want to return 100 documents, but I only have 20 high quality documents.
2:26:59
Sorry, I'm going to give you 80 low quality documents, but we still order them.
2:27:05
The 20 is going to be at the top. So if you want more then you can go to low quality.
2:27:09
We see this over and over in any search at all. The more you scroll and go to the bottom right 100th page of something.
2:27:12
One how this magical thing that the first page missed. Okay, they already know like what they give you, but you are curious.
2:27:18
You can keep on going and then your mileage may vary as you go to the 20th.
2:27:24
You know, for 100 pages it might not be great anymore, but you might find some some very cool page that you wanted.
2:27:28
Okay, but most people don't care about that. All right.
2:27:33
So that's where all of this says really? So you need a two tier index okay.
2:27:37
You have like basically a good index and a bad index so to speak. And so be it.
2:27:40
Yeah. All kinds of ad hoc things that Google came up with over these years okay.
2:27:44
Okay. So query processing how do you reverse engineer. This is a fascinating this is a whole industry.
2:27:48
It's a multibillion dollar industry. The reason is advertisers.
2:27:54
Any anything on this table.
2:27:59
In fact, I won't tell you almost anything in this room right now unless somebody has an apple or a banana or an orange in their backpack.
2:28:01
100% of things in this room right now or manufactured somewhere.
2:28:10
The artificially made. Look at this table. Okay. There's not one natural thing you said you in this paper was manufactured.
2:28:14

Okay? Everything is manufactured object. It means somebody had to sell it.
2:28:21
Somebody had to buy it, right? That's what it's all about. So advertisers are constantly worried that Google is making changes to the search engine.
2:28:24
And when somebody types for white chalk, then Crayola worries that Crayola brand won't come up.
2:28:33
So there's a bunch of companies like those too, especially Search Matrix,
2:28:38
whose job is to have clients that include Coca-Cola, Crayola, Netflix, anybody where the search engine,
2:28:42
the employees that spend all their time literally and for years at a time,
2:28:49
the loss of knowledge in that trying to reverse engineer what Google did to the search engine.
2:28:53
Meaning, how do you even keep track of that? Because those two companies, they have a list of maybe 500 keywords that they, you know, pick by hand.
2:29:00
Some are single words, some of race queries, you know, which they literally search every day.
2:29:08
They get up in the morning and automatically send a search the same 500 words, okay, one at a time and come back with the results.
2:29:12
Right. Look at how the results change over time, like how you did for your homework.
2:29:18
But you compared many different search engines here. Same search engine day by day by day.
2:29:22
Heart. As a ranking between all these different, you know, URLs change or maybe after like one month, some URLs completely dropped out.
2:29:27
What the [INAUDIBLE]? Why did they drop it? Then they are trying to guess what the algorithm is doing.
2:29:34
Okay. And then they publish basically reports for clients that are a lot of money.
2:29:37
And what would the clients do? Read that and know wow, they made that change.
2:29:42
So I now have to make this change to looking more at schema.org, you know tags.
2:29:47
So now my page has to have schema that or tags. So I'm losing out stuff like that okay.
2:29:52
So that's a form of intelligence from the outside. There have no contacts with Google employees.
2:29:56
Don't talk to them. They don't pay anybody okay. Purely as a black box.
2:30:01
By doing the same search every single day, it is actually possible to reverse engineer or Google does so.
2:30:05
Two top companies that do it the most.com, which is not at all the same as De.
2:30:10
Okay, early on I showed you demos. Moss. There is simply an open directory attempt that failed.
2:30:15
Okay. match.com didn't fail. You know they look highly successful.
2:30:19
And the competition company called search metrics, they both do the same thing.

2:30:22
But they're very different companies. I'll bring them up right now okay.
2:30:26
The goal of both companies is the same, which is to find out how the search engine works.
2:30:30
Look at right away content marketing blog. It's all about content.
2:30:36
It's all about marketing. SEO stands for Search Engine Optimization.
2:30:40
OtherWords, SEO is a collection of tips and tricks, heuristics for people that put up web pages that could be individuals that could be small,
2:30:45
you know, home cooks, that could be PepsiCo minnow. But it's basically people that make the content we all consume.
2:30:55
How can they reverse engineer?
2:31:01
How can they architect the page in such a way that when you search for something, the rankings go up to a certain point?
2:31:02
When I have pizza right now, no matter how somebody does search engine optimization at Pizza Hut, Pizza Hut might not come up right away.
2:31:09
Let's actually look. Okay, actually how funny. In this case it did.
2:31:18
I'll tell you why it came up okay. Okay, so maybe a different example. Blaze pizza did not come up at the very top.
2:31:21
So say your employees at Blaze Pizza and they have made the page amazing, right?
2:31:27
The hope will come to the top. It's not about hoping.
2:31:31
What actually happened was that word pizza was auctioned off by Google to all the pizza companies and literally the highest bidder.
2:31:33
The people that paid most money to Google.
2:31:41
Google will make sure that if you type toward pizza, there are, of course at the very top they're selling keywords.
2:31:43
They hard to make $200 billion a year. Okay, so they make these pizza companies fight with each other.
2:31:48
So Italian food type whatever I want okay. What a model. So then that ranking nobody can change because it's literally the highest money.
2:31:53
They'll give you a big good score. Your value would go up like crazy okay.
2:32:00
But within reason, the ones at the bottom okay. Here skip.
2:32:03
Like some of these are all pay. Pizza hut paid the most money to Google, followed by who?
2:32:07
I don't even know. Uh, Domino's. Truly no idea.
2:32:12
Okay. Uh, Grubhub. That's just the delivery side.
2:32:16
Um, yeah. I mean, look at how much Pizza Hut is in.

2:32:20
Well, wait. I'm so sorry. Okay. It's crazy. Start over.
2:32:23
I never said the word. Sorry. I take everything back, I said.
2:32:28
Blazing speeds as high and then pizza place places high.
2:32:32
Then in a pair of pizza. It's cool. Right? Pizza hut literally number four, right?
2:32:36
Because maybe they didn't pay you as much money as the other guys did. That we cannot change.
2:32:40
Okay. Because it's determined by Google is automatic. The highest bidder would have a high number.
2:32:44
Goes up. Okay. But at the bottom somewhere, it might be possible that, you know, maybe this site versus that site,
2:32:49
they can do something in that page so that this can actually go over. That is possible okay.
2:32:57
That is called search engine optimization. You know, that's a game that's been played ever since search engines came along.
2:33:01
I remember in 1990s SEO was a thing. Okay. So SEO tricks, I mean, just Google AdWords SEO tricks.
2:33:06
I will say one more time. So that is an old little thing right there.
2:33:12
20 SEO tips better than backlinks, ultimate tool and smarter.
2:33:17
So over and over and over. Yeah. And these are free tips right?
2:33:21
So if you pay those companies money share like more secrets with you okay.
2:33:25
And it's a cat and mouse game, you know, meaning that you do something weird basically trick tension.
2:33:29
Then Google actually figures out what you're doing, going to meta tag, and you put in the word pizza a billion times.
2:33:34
Okay, the next time Google will then punish you. If you stuff keywords in the meta tag, your ranking will go down.
2:33:39
It's a cops and robbers game, you know? And then the good guys are always one step ahead of the bad guys.
2:33:46
Okay. It's very interesting how this all played. So that's what I'm trying to tell you here.
2:33:50
Search matrix and mods are two companies that do, uh, that help companies rank score.
2:33:55
See there's thousands of keywords, content each page ranking.
2:34:00
Yeah. So those thousands of keywords are simply proprietary.
2:34:05
You know, like I said, they have a list that they want to share with us. Likewise mods also a completely separate company.
2:34:08
So I showed you a search matrix here, right. What about Monster.com?
2:34:14
Same, but it's a competing company. Otherwise it's a monopoly, right?
2:34:18

See that? Higher rankings, quality traffic measurements, all about that.
2:34:22
What companies promise a client or a business owner? We promise you higher rankings okay.
2:34:26
And yeah, they definitely deliver. It's not vaporware by the way. It actually works okay.
2:34:30
Com see all these companies you know, go with these people.
2:34:34
So likewise you know if you go back to you know the other one search metrics they'll probably claim very similar.
2:34:38
There's even SEO conferences by the way. It's completely crazy okay.
2:34:44
I mean, SEO is a very big deal, but we shouldn't worry too much because we don't own businesses.
2:34:47
Okay? But if you do, at some point you do worry about it. All right.
2:34:51
So Search Matrix actually has this nice free document I'm going to show you where they actually list what they think are Google's factors.
2:34:55
What what they call uh, signals that contribute to a good net score.
2:35:04
Okay. So what metrics. In other words, what does Google look in your page.
2:35:09
What what does it look for to make you. Be a higher rank.
2:35:13
Sorry. This one. Yeah.
2:35:18
Check this out. Oh. Rebooting ranking factor assistant replaces.
2:35:21
Okay. And so then what is so cool is this actually yeah. Search metrics right there.
2:35:25
I won't read all of them for you. You can read this afterwards. And they break all the factors down into content based user based technical base.
2:35:29
Even this, if you use a steady base as opposed to rise deep, you will have a higher quality.
2:35:37
Okay. And so on. Um, file size.
2:35:44
So if your images are pretty easy to load, well, you'll go up high in quality.
2:35:47
If you punished user making them spin right then that's a bad thing.
2:35:52
Okay. So these are all common sense obvious things. But it's actually such a neat thing.
2:35:56
So any of us even URL length to make a massively long URL that nobody cares to remember will punish you.
2:36:00
Okay, that's why Bitly you are shortener is a popular because smaller links are easy to remember.
2:36:06
And so then in terms of the ranking, Google will actually, you know, put you at the top.
2:36:11
Okay. Wow. So so many things that you had no idea about mobile friendliness.
2:36:15
Right. Again, if you make your site pretty easy to load on like standard fonts and things,

2:36:19
right, then it will actually go up if you make your site be voice friendly again a lot.
2:36:23
Uh, conversely, you'll be punished for all the things. And you look at this user experience, even number of images, video integration, font size.
2:36:29
Yeah. You know, so many people, you know, when you get old you cannot read tiny font.
2:36:36
Right? So if you have pages, very tiny text go down and ranking, you know, because people hate them.
2:36:40
So like common sense in a way. But it's actually very cool.
2:36:45
Um, yeah. Unordered. That's really funny. Unordered list.
2:36:49
I don't know if that's a good thing or a bad thing. Bullet items. Right? Yeah.
2:36:53
Bullets per list. Suppose you make a bullet list with, you know, 200 items.
2:36:56
You will. It's painful to scroll through all of them screen and probably only look at the top ten anyway, so keep the bullet list.
2:37:00
Very small number of items, very small. So many.
2:37:05
Okay. Like with social signals you know. So they actually have links to all the social media.
2:37:09
If you do let's say page rank. Look at the whole backlink stuff okay.
2:37:14
So there's so much in here I won't read. Every single one of them is actually described in here.
2:37:17
So to me this is very cool. You know juicy document. We will not test you in the exam for any of these.
2:37:23
You know any better. Okay. Okay.
2:37:30
Again. Most. Yeah. Then. Exactly. That's what they do, right?
2:37:34
They simply search for the same terms over and over and then see how the thing is changing.
2:37:38
And how many times has Google changed the ranking? Many, many, many times.
2:37:42
Google actually has a page, which I think I have a link for,
2:37:46
where they tell you officially every single change that I ever made to the extent, without revealing too much secrets.
2:37:48
Okay. So that it's not all entirely like a black box thing, you know?
2:37:54
So they have a page that literally going back to 1998, every year that changes are made that they can make public.
2:37:58
So rich history of how search works actually. So if you then, you know go to Monster.com and become a customer.
2:38:04
Here's what they could do.
2:38:11
They will help you in one way by going to your page Pizza hut.com/in all the right or menu and then rank you in relation to your competition and say,

2:38:13
see, this way you're not getting customers because you said that's highly useful, right?
2:38:23
And they also have this most casting where they internally tracking whatever
2:38:27
changes happening and then communicating that all to the customers not publicly,
2:38:32
but you know, privately you got to pay them money. And so then in other words, this is specific to you.
2:38:36
This is not specific to you. It's for everybody. But to help all the clients call for two different ways to make money, okay.
2:38:41
Yeah. Look at this. Whoa. This is crazy, right? So Monster.com also has a page like Google has a page.
2:38:47
Wow. Okay, nine years worth of algorithm update. Go on.
2:38:54
Go through all of them 2014 to 2020. Right. Look how big this pages.
2:38:57
See that? Okay. Sites disappear. You know, releases like all of this and or releases, you can click on every single one of them.
2:39:02
It goes to google.com somewhere. See that? So actually go search engine London.
2:39:09
Yeah. So we have no time to read. We actually don't care about all this.
2:39:14
But it's a very, very big deal. You know it's a pretty big deal. By the way, recently something happened this past weekend.
2:39:18
Actually, it's like a little alarming thing. Some events were watched by random YouTube watchers like you and me.
2:39:25
And so the government, the US government. One of the list of who watch certain videos from Google.
2:39:32
My God. Okay. This is the United States, not North Korea.
2:39:40
It's not China. It's not Iran. It's not Russia. But that's actually what the government wanted.
2:39:44
Okay. They're interested in seeing who watch those U.N. videos and so called and push back.
2:39:49
They're not obligated to endanger them. But then the government can have more forceful ones and say you have to under lock your doors.
2:39:55
So that's a very scary thing actually. Okay. So it means you might be watching something completely unrelated to anything.
2:40:01
Um, Harikrishna, you know, like dance party in Venice Beach. Okay.
2:40:07
But in the future, somebody might come and decide, know criminal. So you know.
2:40:10
Fine. Tell me the people that watch that video. Holy crap. British care.
2:40:13
Okay, so anyway, this is a very huge, crazy. Look at this.
2:40:18
So big. We have no time. But then if you read you here title tags.
2:40:21

Okay, there's some individual title tags.
2:40:26
So if you carefully read through all of these, you meaning an owner of a business, then you will know so much that your competition does not know.
2:40:28
All right. So much rank. Again that is how they take your site.
2:40:36
So you are the business owner and give you a ten point measurement called the mass rank.
2:40:40
You know, and then it's like page rank. But from the outside and in relation to your competition to the most rank is pretty high.
2:40:44
You don't have to worry about the competition and vice versa okay. All right.
2:40:51
So then how are hardest Moss rank work? Again, a lot like page rank. See this?
2:40:55
So how many people are linking to you. That's page rank versus how many that you are supporting right.
2:40:59
And then trusted sites. Cool. So if National Geographic points to you your quality is going to go and independent of anything else,
2:41:04
if you have high quality that the user the world wants not BS crap,
2:41:10
okay, obviously you're going to be rewarded by that and social signals and again like social media linking to you okay.
2:41:14
So then that is called a moss explorer where you can put in like any URL you want, you can go and look at all this afterwards.
2:41:21
You can see this. They even tell you like how they indexed or you know, okay.
2:41:26
So for all this to happen, there are also like a search engine right to go and index to so many people do indexing by the way so much.com.
2:41:30
Also this this would be like your homework by the way. It's like so cool.
2:41:37
See there's an index crawler all of this called dot but.
2:41:41
Oh, okay. Here's a question for you.
2:41:45
You think DNS is cool, right?
2:41:51
When you have an IP address, you know eventually that number will become like a name by domain name service lookup, right?
2:41:53
What is an alternative to DNS? DNS is not the only thing. Does anybody know?
2:41:59
I won't even answer you. I'll see if somebody knows. Kind of look for it.
2:42:04
It's got the letter T in it. An acronym.
2:42:09
Okay. Wow. It's almost like an alternative to DNS.
2:42:12
No, there's an alternative. All right, so then, you know, we'll take a break for the last 20 minutes.

2:42:17
Take a five minute break. Okay. Sorry. I have a lot to say, you guys. You know, I just have to finish it.
2:42:22
Yes. So in the end, that's all it is, right? So mods has these hand-picked keywords that they came up with and every day the search and
2:42:27
then see how the search goes up and down in terms of rankings what terms get dropped.
2:42:34
You know, and some that they can basically guess like what change got into the search engine.
2:42:39
Okay. So the Q delta between day by day for for those thousand keywords okay.
2:42:44
And then some kind of temperature. Uh, interesting. Yeah.
2:42:50
So there's some kind of a, you know, top ten actually very interesting.
2:42:53
Right. Of these thousand keywords, you know, you search for them and then for each one of them take the top ten results that come back and then
2:42:56
only compare the top ten day by day by day and turn that into what they call a temperature volume.
2:43:04
So I don't know the details for all of these, but it doesn't matter. You know what I started?
2:43:09
Com just does it for a living. Okay.
2:43:12
We can then slowly start to finish all this up by going to the last part of what we promised, which is if you go to site, the page one.
2:43:13
Right. We said there are three parts to all of this.
2:43:20
One is you can do changes on the inverted index, mostly coming up with the goodness term okay, the G term.
2:43:23
And here we said there are two companies but many more that pretty much search Google all day long.
2:43:29
And some. The results reverse engineer and maintain like lists for their clients.
2:43:35
And finally the third one is we can look at Google's internal architecture, which also completely changed back.
2:43:39
You know, I mean from back, what used to be it's not all about PageRank anymore.
2:43:44
It's not all about also, uh, MapReduce anymore.
2:43:49
There's something called hyperscale, right? Likewise, PageRank is now being replaced by something called Rankbrain.
2:43:54
In other words, they learn from their own, like, you know, mistakes in a way, right?
2:44:00
Like what didn't work, where your page rank is, all that there is. It's not all that there is.
2:44:03
So there is this cool thing called how Search Works the story.
2:44:06
It's very cool, by the way, lots of cool videos in our text. You can watch.
2:44:11
Okay, okay. But you you definitely know all this at this point in this course.

2:44:14
Yeah. Look at this. Wow. Did you mean showed up in 2001?
2:44:19
You typed something, and then they fix your typo, you know, say Gorbachev.
2:44:23
You typed Gorbachev with the typo that said, you mean Mikhail Gorbachev?
2:44:27
You say yes, right? Because you want the right person. And then synonyms, you know, they have this lemma,
2:44:31
kind of a dictionary scheme where you type a word and then they can look for synonyms and then stock quotes.
2:44:36
You type AAPL give you the Apple stock quote. Autocomplete showed up as early as 2005.
2:44:42
This very cool video flight data. If you put in like an airline's number, it will tell you like you know, the landing time, takeoff time, right?
2:44:47
And then movie times. That is very cool. You put in the name of a movie in Italian, movie times and patent data.
2:44:54
These are two separate things. But, you know, and then mobile apps that came in because iPhone was introduced in what, 2007.
2:45:00
So Steve Jobs, the life one. So right the very next year, you know, search mobile app and then voice search so slowly.
2:45:07
Now Siri, you know like all those you know okay home order right. And then Google instant image search knowledge graph internally.
2:45:13
We didn't know that. And then carousels you know carousel is where you have two arrows okay.
2:45:20
And go click click click spinning the carousel one way a click click the other way.
2:45:24
By the way our course is a carousel. You know that's a carousel. Look forwards 1920 and 2122 way up to 32.
2:45:28
I can also go backwards which is click here. Go backwards.
2:45:37
We spin, spin, spin, spin counterclockwise and then you can spin.
2:45:42
We go this way. I have a doubly linked circular list okay.
2:45:46
That's actually what it is. So then many searches actually appear that way if you didn't notice.
2:45:49
So that is all I'm going to talk about very quickly the whole Rankbrain stuff okay.
2:45:54
Hey, you tell me. Should we take a break? Yes or no?
2:46:01
Yeah. Who says yes? Put it behind. Are a few of you.
2:46:05
I think you are, like, basically outnumbered. People don't want to break up. Get it over with.
2:46:10
Tell me. Okay. I'm going to tell you. We'll stop at 810 or something.
2:46:14
We're going up. Okay, a bonus. Okay. So, uh.
2:46:17
Yes. Google. Yes. How the search changed over the years.

2:46:23
There's a lot of changes that they've made, obviously. Right. And yes, so 1999 was so simple.
2:46:26
See this? That is the main thing that they have indexed servers.
2:46:33
The whole reverse the inverted index okay. And those index in turn, you know.
2:46:37
Uh okay doc. Yeah. So that this must be, I guess, cached documents I think.
2:46:41
But this the main thing that they use the index servers. Okay. And obviously there was some kind of a scheme that spread into all of this.
2:46:47
Okay. And they had an ad system that they figured out okay. That was very basic okay.
2:46:53
Back then. Right. But now, um, yeah.
2:46:57
Oh, wow. This actually crazy when Google started the whole site, the whole site architecture was a Stanford University css.sanford.edu.
2:47:01
That is what this from an early diagram going back to 1998 or something.
2:47:10
By the way, the search engine version was called BackRub. Okay. Actually named the search engine BackRub.
2:47:15
BackRub. Yep. And it's so simple. Look at this.
2:47:20
This is really a homework a course homework okay. See there's a crawler that goes crawling right.
2:47:24
And then grabs pages like from all over the world, puts it in a repository and then indexes it.
2:47:29
Okay. So the indexing is the key. And then it's all the indexes are sitting in all these barrels okay.
2:47:34
Like right there document index. And then there's also a page ranking going on and the searcher.
2:47:40
So then the user will actually finally use etc. researchers getting results from all these index belts okay.
2:47:45
So very simple architecture. If you want to know how every piece works please go to the site.
2:47:51
The site still exists in for Lamda Stanford you the site does not exist.
2:47:55
Go to archive.org and type that will actually come up. Okay.
2:47:59
But let's see. Let's see how this changed. All right.
2:48:04
So the query query means something that I'm typing for like for example the word ranking factors.
2:48:07
So what happens when you type something. So word IDs that means remove the stopwords and actually make them high quality words.
2:48:12
And look in the inverted index. And then find all the documents that contain the words that we put in, like for example ranking factors.
2:48:17
So for you understand fully right. It's all scanned through all the documents, contains all the search terms.
2:48:25

Again we said when I have multiple terms, look for documents that have all of them the secret and between all the words okay.
2:48:31
And then compute the rank. So page rank. So this means, you know for for giving it back to you because already the page has a page rank.
2:48:37
But now you can add a good net score and all of the page has it added all.
2:48:45
And then for the user compute the rank. Meaning I want to give you 100 documents.
2:48:49
In what order should I give you 100 documents? That is what is not page rank that link.
2:48:52
Okay. All right. So then yeah, keep doing that until no documents.
2:48:56
Meaning you search through all the indexes and find every single document you can potentially serve and return the top care of them.
2:48:59
You know, that is actually what is happening.
2:49:06
I'll just say you in Google ranking factors, you will find documents with the word Google ranking and factors and do everything we just talked about.
2:49:09
And here it is within like you know less than 0.3 seconds.
2:49:16
Pretty amazing right? So that is what uh mostly happens.
2:49:20
And then what else can we do though. It's more than keyword matching.
2:49:24
Ha. This is basically machine learning. So gradually the notion of semantic search you should not be limited by just a keyword that you put in.
2:49:27
Keywords are cool. But what about documents? They don't even have my keyword, but still related to like what I'm searching for.
2:49:35
You need the meaning. Syntax means grammar. Semantics means meaning.
2:49:41
Tim Berners-Lee. Yesterday.
2:49:46
Is the invoice published? I got 30 or, you know, kind of a summary of the webinar where he's saying something interesting, but he's an idealist, okay?
2:49:50
He's not gonna succeed. He said the web has been entirely hijacked by data people.
2:49:59
Our data is being bought and sold, and social media companies completely took it over and screwed everything up.
2:50:03
Chase has to go back to a more pure form of the web, where we will be owners of our own data.
2:50:09
Okay, so neat document, but it's very sad to read because you need to be highly technical for all that to work.
2:50:14
The whole world is not highly technical. They don't give a rat's ass. So sadly what he wants will not happen.
2:50:20
Okay, I will find the document also for you. So a bunch of things to do and get back.
2:50:25

Find you cool things to read. All right. So anyway, he says, um, Tim Berners-Lee has always wanted the web to be semantic.
2:50:29
Okay. When he made the web, by the way, he is not the person who said search engines have to be keyword based, whereas Google idea.
2:50:36
Okay. And it one I mean it's a good idea. But he never wanted that from day one.
2:50:44
His version of the web search, Tim Berners-Lee version of the web was all meaning based.
2:50:49
Okay, but he never got it. Okay. Yeah, that's what he said. Yeah.
2:50:54
Cool. Uh, yes. So then the whole search, query processing, you know, this is what we looked at, the whole relevance scoring and then search results.
2:50:57
You know this like search placement ranking placement.
2:51:04
All right. So semantic ontology you know.
2:51:07
So this all just simply means again what are they do now when you type something they try to guess the word avatar.
2:51:10
They try to guess that it's a movie and suddenly they can give you avatar movie documents.
2:51:16
If they know I'm searching for Avatar Sanskrit word, they go to Sanskrit grammar.
2:51:19
Start giving me Avatar Sanskrit meanings, right? That is called genre ontology identification.
2:51:23
And then they can look for entities, right? Named entities, you know, already know that.
2:51:29
And then they try to understand your whole phrase. That's all ML gradually this the hardest of all really.
2:51:33
What am I looking for? What is meant and why did I typed it the right way? Hard.
2:51:39
And then semantic annotation again meaning based and then go find the results.
2:51:42
Okay, so gradually the world is moving more towards semantic stuff, you know, including limbs okay.
2:51:46
All right. Again, same thing. Right term, which is actually the word that you put in versus entity, which is the thing you're looking for.
2:51:53
So red stoplight, you know when it's a red stoplight, what if you look for the words red and stoplight or actually look for red stoplight pictures.
2:52:00
Because you know what that is. Those are two very different things okay.
2:52:08
In other words, I can have a blog about red spotlights.
2:52:12
I can have a picture of every single great stoplight that I ever took, but I never say the words red stoplight anywhere in my blog.
2:52:14
In today's Google, my page won't be shown to anybody because I never said the words red stoplight.
2:52:21
But if you do meaning based search like that entity based search, all the red that lights up in a stoplight will all go to that

part.
2:52:26
All the greens, all the oranges in Japan, by the way, there's also one more color blue.
2:52:34
Japanese stoplights are blue. Okay. You know that, right? Yeah. Okay.
2:52:39
Then that can go somewhere over here and you can do searches based on that.
2:52:43
Okay. Very cool. Right. All right?
2:52:46
Yep. Okay. In other words, there's a thing called stop.
2:52:49
Light is an entity. It is not simply a combination of the words is what we're trying to say.
2:52:54
It's the whole thing. But so okay, so what does the modern web do? Well, financially.
2:52:58
There's no need for too many, um, details, you guys. Okay, the big two buckets are here.
2:53:02
Right there. This is the classic syntax based search.
2:53:07
Keywords search. Purely grammatical. No, meaning nothing here.
2:53:11
The opposite. Purely AI, knowledge based.
2:53:15
Meaning based, not keyword based. Salaam Gemini barred all the right open.
2:53:19
You know a jeopardy. You know all that is all.
2:53:26
And this part is slowly going more and more towards the AI part.
2:53:29
Okay, this whole idea, this will always be useful, there's no doubt at all, but less and less so.
2:53:33
And then your knowledge your gone through all this okay. So knowledge graph they have their own big operational knowledge graph.
2:53:40
When you type something they can find what you type in the knowledge graph.
2:53:45
And then follow the links from what you type to the nodes nearby and summarize all of them for you.
2:53:48
When you type someone's name, I'll type Elon Musk, I'll type Steve Jobs, and then that name will then go to a knowledge graph node about him.
2:53:53
And then [INAUDIBLE] tell you, you know, founded Apple Corporation. All this. Right.
2:54:01
All these came from a node graph. There's no page already with all the information in it okay.
2:54:05
Because I can assemble it at runtime. So Knowledge Graph will come like super useful and all that okay okay.
2:54:10
And then your examples Oliver Bloom I don't just type people's names.
2:54:16
You'll see what happens okay. Yeah. You see so Volkswagen bus right.
2:54:20
We never said all that. But it knew from the knowledge graph that person has an entity called uh CEO of in a VW.

2:54:24
And people want to know that. So actually I always love knowledge graphs.
2:54:30
And I've told you many times it is the error that humans make. So we put our knowledge in the graph and then it spits it back to us.
2:54:34
And they always like a little bit somewhat dislike, hey,
2:54:39
there's too strong word machine learning because that is basically a big sledgehammer that you smack the world with.
2:54:42
Okay. Just patterns and data and it doesn't work very well.
2:54:47
Knowledge graphs always work. And so it's a Google yes knowledge graph. And then this notion of a rank brain okay.
2:54:50
So then rank brain harm okay. See this is very interesting right after the initial subset meaning you get the K documents right.
2:54:56
How do you rank them. You know. And how do you serve them. That is why this idea called the rank brain actually kicks in.
2:55:03
It started in 2015 by the way. So keywords become entities.
2:55:09
You know, that's I mean there is no magic here. Any word that you put in there try to make it.
2:55:13
An entity like Steve Jobs actually became like an entity in the Knowledge Graph.
2:55:18
And then they can serve you higher quality result that mapping they call rank brain necessarily all okay.
2:55:22
So rank brain is like a lookup almost that takes like raw words and tries to
2:55:27
make them into knowledge graph nodes because then you get high quality results.
2:55:31
Okay. That is all cool. That's what all of this is.
2:55:34
Okay. So new search results. You can read all this afterwards.
2:55:37
Wow okay so links you know and words rank brain is a pretty high factor in how they rank like what degree and what order.
2:55:41
So very important algorithm for them. Okay. Like this site?
2:55:51
Which country has the best course identifier?
2:55:56
And then this is all about some details about how Rankbrain works. And I'm going to skip it.
2:55:59
All right. No not really. But the idea is novel queries.
2:56:03
You know, like like that that that query has never been asked. Okay. But what Rankbrain is able to do is go ahead and word by word,
2:56:07
and then have words like country cars and then give meaning to those words and find some kind of entity that matches that query.
2:56:14

Exactly. And then answer the query. So ultimately what it does.
2:56:21
So therefore. And also Rankbrain is constantly changing, right?
2:56:26
Because what people search for constantly changes. Basically, every search can add more data to this deep learning algorithm.
2:56:30
So over time it will get better and better. None of these get worse over time, not only get better over time.
2:56:36
So you can read about Rankbrain if you want and then um, yeah, again, that's the main like right there.
2:56:41
Okay, so this actually very rankbrain turns keywords into concepts like entities and looks them up.
2:56:48
Right. But then they put the results up in a certain order for you and they watch you, not in real time by the log.
2:56:54
They can tell whatever to put at the very top. Did you actually click on it?
2:57:00
The answer was yes. Then rankbrain work correctly in the whole mapping between keywords and entities.
2:57:05
If they put up five links, right? And many people click on the first link.
2:57:11
It means Rankbrain is basically screwing up. Five should be number one.
2:57:15
So then they take your own behavior.
2:57:19
When you say you, meaning millions of people's behavior and modify their own algorithm like a form of feedback okay.
2:57:21
Reinforcement. Feedback. That's very cool. So then that is what all of this is.
2:57:27
Okay. So then they use these user signals like what we do with the results they give us and say like this up rank that page.
2:57:31
So if they give you things in a certain order and then you actually click on it, it means yes, the page rank actually work, page brain actually work.
2:57:38
So then make the rank even higher next time. Conversely, punish it know if the user does not click on it.
2:57:45
Okay, so user from us. I'm going to type this.
2:57:51
I'm going to write this. Human feedback in Transformers and ChatGPT.
2:57:55
Actually, why it works is because when you take the raw transformer that it has and you put in a query,
2:58:03
it'll give you horribly bad things that nobody in the world wants to redo, and it won't be usable things.
2:58:10
Okay. But it doesn't do that. It gives you mostly good answers, right?
2:58:14
Because humans, we used to basically punish the algorithm when it gave you bad output, right?
2:58:17
That is called human feedback. So likewise here every user's behavior when they give you position a certain rank,
2:58:23
and whether you obey the rank or you skip the rank and do something else is a form of human feedback.
2:58:30

So we all provide human feedback just by using the search engine.
2:58:35
And they in turn will then modify how the search works. Okay. Meaning how they ranked algorithms.
2:58:39
That is all you guys. So if you want to know even more about page right.
2:58:43
And the rankbrain right there, they have a pattern for it. Rank brain patterns.
2:58:47
You can go and look at it. I'm purposely not spending time on all this okay.
2:58:50
Yeah. See this actually very interesting right. Past searches.
2:58:55
And then learns by matching search results. That means what order the results represented, what other people clicked on it.
2:58:58
Then it this association quote. Um, and then stopwords.
2:59:04
Okay. This is actually very cool, right? Rank brain.
2:59:10
Well, not unlike the previous search that ignored Stopwords.
2:59:14
This one will not ignore Stopwords Stopwords are not bad.
2:59:18
Flights to London is so different from flights from London.
2:59:22
So you're in L.A., right? Flies to London is so different. I need to get a British passport in and go to Heathrow airport.
2:59:27
Flies from London. I want to know what time I have to go to the word from and to become like, damn important.
2:59:33
So Rankbrain thankfully takes that into account. Okay. Likewise, Rankbrain is obviously smart enough to know like vernacular.
2:59:39
So when you say the word boot, boot means something that you put on your, you know, your feet, right?
2:59:46
In England, what boot means the trunk of your car. So then it uses your location.
2:59:51
Now if you are searching in UK or in the US and that's the right thing with the word right, it's not a big deal.
2:59:55
But it's important. So they do that as well. Okay. And last slide.
3:00:00
And we can actually stop I promise. Um yeah.
3:00:05
So offline sources you know this actually means once again you have this knowledge graph right.
3:00:08
Which is offline. So they can use the knowledge graph and constantly make associations between what
3:00:13
people searched for and what people could be served when they search for something.
3:00:18
So then once you make the association, if you search for Tokyo, give you results about Japan and vice versa.
3:00:22
Just constantly learning is the point, okay? Endlessly just learn.
3:00:28
Make more and more associations in our newsroom in the future. Sorry, this was a lot.

3:00:32
And then I'm still back by one. Uh, I'm behind by like one. One topic and I will catch up.
3:00:37
But you have went through, like, a lot, right? I didn't wanna, you know, I mean, just four minutes and I said, I'll stop at, uh, 810.
3:00:41
But this is funny here. So many of our corrections, at least, if nothing else, go play with this.
3:00:49
Okay. Edit distance. Go to this one slide and have fun with it.
3:00:55
Like here. I have to show you. That that.
3:01:00
Um. Yeah. That is. So go to slide number 2727 and type words okay.
3:01:05
Let us type. Let us type USC and then turn that into UCLA.
3:01:14
Okay. So the whole spelling correction algorithm tries to find out from one word
3:01:20
what is the smallest number of changes you can make to get to the second word.
3:01:26
It does it letter by letter by letter meaning it says for the first letter and first letter.
3:01:30
What change should I make? Nothing. Because you is already you.
3:01:35
Let's call source is called target. Right? Next second letter.
3:01:39
Second letter. What change should I make? I should change s to see.
3:01:42
Cool. Let's want to change. Okay. Then third letter I should change C to L.
3:01:46
That's my second change. First change was S became C and then now C we came out.
3:01:52
Last changes I have to add a USA does not have a right.
3:01:58
So the whole algorithm works by finding out what to change, what to add or even what delete.
3:02:02
If I had UCLA first and USC afterwards, it will say well, what do I do with a there is no add, then delete.
3:02:08
So you can modify which is substitute or add at the end or delete at the end.
3:02:15
Those are three things. So watch this okay. Cool.
3:02:20
Those are the three changes, right? So the first change is between 1 and 1.
3:02:25
First letter. Your first letter you know change between SNC and change STC.
3:02:29
So one one change at the diagonal one change.
3:02:36
And afterwards I had to change C to L. So then second change.
3:02:39
And finally I had to add a. That's my third change. So this monotonically increasing number of changes is called edit distance.
3:02:44

How many total changes were made to go from USD directly?
3:02:52
Three changes right. This number will only go on increasing because in some letters is no change, so the count stays the same when a change is made.
3:02:56
Plus one plus one plus one. That's all okay. Meaning will go.
3:03:04
When I do this we'll say UCLA, USC exact same number of changes which is three.
3:03:08
But now USC is going to go here. So let's continue here. So then this matrix will be like a little bit different in dimension.
3:03:13
That is all in shape. But you going to get the exact same right number of changes.
3:03:20
Watch this okay. See the same trick. But now the shape changed obviously.
3:03:24
And why three. Again because you does not have to change Samuel.
3:03:29
She has to change the s. That is one change L has to change the C.
3:03:33
That is the second change and it has to be dropped. That is the third change.
3:03:37
That is all Levenstein distances okay. Variation. Amazing algorithm is one of the coolest algorithms in the world.
3:03:41
Relevant tell you so have fun exploring this.
3:03:46
That is all. So what should we do before we meet? Hopefully you will have the next homework.
3:03:49
When? Not before. Wow. So have fun. Okay, bye.
3:03:54
You guys, it's exactly 20.
3:03:57

Lecture - 3a
End of action. Right.
Some hard candy here. Okay. Let us start.
As usual, I have a pile of things to tell you. It passes.
It is. And.
Cool. Oh. We'll.
High level. First things first.
Homework. It's not due for three more days. One more piece of advice.
If you have not done the homework, take my proposal. Still married, pillock, Just take it.
Okay. And there's two ways to steal material. One, you can go on to replicate.
Come and get a free account and just a fork. You will get all my files.
It has all the jar files you need. When you run, you will actually see the word count.
Okay. You can trivially modify word count into the homework.
It is probably the easiest thing you can do and grab the java files and just submit.
You can download them again or you can take the whole wrapper that I have with all the files in it and download a zip copy like a local,
you know, installation on your laptop, and then you can double click on the zip.
You'll get a directory. This is homework three and then you have Java already installed in your system, right?
The command line you then that oh, you can do eclipse. So some kind of idea.
I use something called Doctor Java by the way, if you have no idea what the heck Doctor Java is,
if you want to learn Java and you want to keep it very simple, Eclipse is too complicated.
You don't need eclipse. Dr. Java is from Rice University and it's been written in Java.
So download if you want doctor Java and do your homework in Dr. Java and if you like scalar, there's a different programing language here.
You can even get Dutch Java. Yeah, you know, that's an easy idea.
So one way or the other, you can do your homework easy. You still have three days, your last homework.
There's two more homeworks. You go out, there's only four more weeks. So I'm going to do something interesting.
I'm going to slip five and four. Just pretend they didn't read it. Okay?
Okay. Because it's invalid indexing after all. I'm going to read it and I'll make five, which is now the four here entirely optional, voluntary.
You don't have to do it. I still give it to you that we will have the homework done.
Please do it after the exam. Okay. It's nice to do it, but it's still just simply inverted indexing.
You're done that okay. But then you will use pretty cool engines, solar engine and then which just written in Java and Lunar.
That is a JavaScript port. So if you have salary, can a lunar Lunar Digest is very neat.
So I'll give it to you. You can actually pay me. That is pretty cool.
But then you don't need to do it for credit because a credit would only involve four homeworks.
You can scale all of them up, you know. You know the scaling doesn't matter, right?
In other words, right now they're worth, you know, ten points each. Imagine they were 12 and a half points each.
It truly doesn't matter. In fact, if I just leave it at ten, if you think you know, that's too bad, let's leave them all at ten.
This is taken out. So the whole exam is now the whole course is now for 90 points, I suppose.
100 points. You know, the course can be for 47.8 points.
Okay. I can still give you a grades. That is the magic of relative grading.
I have no skills that I probably should have nothing in my head that says only 10% of the class will get an error.
It's all relative. So 90 is as good as 100. Just add ten to it.
Okay? Because basically know what carrying would be or multiply nine by ten divided by nine.
Okay, You're going to get it at 100. So it doesn't matter really. But I'm going to make the first homework optional.
Meaning you will only have one more homework and the homework you will have is going to be pretty cool.
I'll add a little Twitter. So what I'm going to do is this last home request your homework number before I'll give it to you.
Right after this is due. She will have like almost three weeks to do it.
It is going to be wonderful because you will use this new kind of database called the Vector database.
She will take questions. Jeopardy questions, Jeopardy Q&A.
There's a database for that 1200 Jeopardy questions. You can do what is called vector raising or embedding.

You can actually in or turn each question and answer into a linear python list of numbers, floating point numbers.
That's called embedding. And then you can do Q&A, you can converse with it.
You can ask Catalan to actually say, you know, tell me about biology or tell me about insects,
and then it'll find your actually the Jeopardy questions related to insects. Okay.
So it's basically amazing. So that is a new way to do search.
It's a non keyword based search. It's semantic search. So absolutely great that homework is fun.
But I'll add one more second part to it. The second part would be also semantic search for you would call it retrieval augmentation.
We're going to call it drag, but this time it'll be based on some PDA file that you have a PDA, if I can be a book,
so we can then embed all the sentences in the book and you can actually chat with tell them to have you explain the book.
You're in a book about object orientation.
You can say, Explain polymorphism, and it'll go in the summary of the book and actually find your polymorphic explained polymorphism.
I suppose I'm, you know, in fourth grade, in 100 words, I mean,
you can really play with lamb and see all the amazing search results you're going to get.
Okay, so that'll be the part before this. So I'm still writing that up and even more amazing, how are you going for it?
And the UI can be then the back end, which is actual calculation.
Let's see what augmentation can be uploaded to your website. You're on GitHub that I own and you will have a nice UI.
Then you can send the UI to anybody in the world and they can then interact with the other chat.
It'll help them. It'll explain your PDF file to them.
So this whole new world offer attitude augmentation,
meaning we take the standard Al-Alam and then we basically make it be like highly high quality answers because the answers come from your data.
The difference here is in this case your data would be JSON question answer pairs like Jeopardy, the game of Jeopardy!
In part B, it it'll be a big PDF file with all kinds of like paragraphs in it.
So it's pretty similar. So the bot call retrieval augmentation, they're called rag.
So Rag Rag Al-Alam, that is going to be the last homework.
IBM has a page on it. Everybody has a baseline. And I think explaining this to you one time, okay, so drag is basically this extra stuff.
So the user goes to an alarm and alarm can answer directly back to the user, but then the answers might be very bad.
They might be wrong. We call it hallucination. So what about this extra stuff?
See that extra stuff? That extra stuff is where you are able to go to a database and actually augment and augment your prompt.
Actually with this extra content you fetch and that becomes a new input to the Dalai Lama.
That is the reason why the answer becomes high quality. Wonderful.
So three things to tell you that are all based on very similar things.
So search is changing completely. Okay. I mean, Google didn't cost is basically opening right through the gantlet and then Google is pulled into it.
So there's no going back. It'll all be this new way. So I have three topics today in a question answering, you know, auto completion.
And then also I think forget the clustering that all change based on lamps and all this anyway.
And so I'm slowly transitioning this course. This is only the second time I'm teaching this class, so I don't have brand new things.
But last time I mentioned Spring already started mentioning all this, but I will mention even more things.
Okay, I'll say some here and say some more here.
I want to explain how Transformers work and I want to explain how alarms work and how to trigger augmentation work.
So in a way, you know, I'm giving it a homework before I explain it. It doesn't matter.
The homework's very simple. I'll give you starter code and then you can actually use it.
Okay. It's based on Python, Python and JavaScript.
But the whole notion of I want to tell you a little bit more. All happened within the last two days.
Heart of the press. Number one, I'm going to go backwards is the most recent one.

So Samsung, you know, obviously a Samsung phones that actually run go up against Apple phones and make high end phones and they tabs,
you know because Samsung also runs like a charger pretty.
And Samsung came up with Wunderlist yesterday and this new alarm large language model runs on just Samsung phones.
It does not run on Samsung server or something.
So the big claim to fame that they have it is privacy focused, meaning any any data that you have on your phone.
It stays on your phone. It does not get uploaded anywhere. So that's called go, goes, so goes.
Is Samsung's brand new alarm two days ago.
Oh, cool. The Verge Samsung, such a PDA rival is coming soon to its devices.
So you can read, read reader and you can understand I like that Samsung unveiled its own model.
So that's going to be neat. Apple also has one. They've not released it fully yet.
Right. Apple will come up with their own generator in a chat. Everybody's going to have their own alarm because that is the way to go.
And then such an alarm can actually, in Apple's case, query things across the entire Apple ecosystem.
You can search for music, search by iTunes apps. You can search for movies, you know, search on your laptop.
Yes. Search this search chapel. The next announcement actually came from Elon Musk.
So Yellen said we saw it on and Sam Altman used to be friends.
Their boss started opening. Right. And then in spite of the name opening, I can't open a is not open.
In fact, we still don't actually know the real algorithm that is used to train such a team and how they use humans to provide feedback,
you know, so like a secret. And that bugs people because you really don't know what is in it.
So Elon basically publicly said, Oh, that sucks. You know, I'm not a friend anymore.
So and then there's like a rival, right? He announced recently, two days ago a new Al-Alam that is called Grok.
It's a claim to fame is it's Elon Musk after all.
You can ask Grok so-called dirty questions.
You can ask it anything you want. Inappropriate, not straightforward questions.
It'll happily answer you, answer you spicy questions.
Ooh, a rebellious air with few, if not no guardrails or no.
Yeah, it outperformed. I mean, you know, he claims, like, all kinds of things, right?
But it's going to be very interesting to see what it's all going to happen.
Just watch it for a second. Lately, one way I've been able to help keep my mental health a priority has been through therapy using better health.
After graduating college and now being,
I'm I'm going to start something which I called Truth Djibouti or Maximum truth seeking a guide that tries to understand the nature of the universe.
And I think this just might be the best path to safety. Okay.
So back in April, Elon Musk said he was going to launch his own artificial intelligence.
And this weekend he did. Yesterday, Elon Musk debuted Brock, the newest A.I. chat bot to hit the market.
His company, X A.I., launched the tech to a select group of X users to test it out.
The company describes Grok as a rebellious streak and that it should see that you know how to make cocaine step by step.
Come out with bat swinging. Go strangle somebody with their bare hands.
Probably answer you seriously. I don't want to say dirty things.
Okay, so I'm almost ready to say it, But imagine whatever the [INAUDIBLE] you want to ask.
Okay? Yeah, I'll explain to you. And, you know, I mean, that's good or bad.
If the kid trash it, the kid would die. Okay. Cocaine is like little for little children.
So it is actually very sad. You know, how do we use acid battery to blow somebody up?
You who knows, right? What people are going to ask. It has no guardrails.
It actually answer you. And if you say a white board, not answer it where that's not opening,
I should answer it because those answers where those questions were asked and
those answers came out and then the chat bot was punished by reinforcement.
Okay. Was basically punished so that it knows what balancers are and one generated anymore as a last layer of tuning.

Okay. It's called human feedback, reinforcement learning for all.
This one does not have a. So this one is a hard data, whatever the [INAUDIBLE] when it's actually spit it out.
So it's like basically a person with no filter between the brain and their mouth, whatever the [INAUDIBLE] they're going to say.
Okay. I mean, that's very scary. Okay.
I mean, seriously, it says start cooking and I hope you don't blow yourself up or get arrested or get ordered and die.
There's also that. Okay. That's very scary.
But there is a lot to say. And then just one more material.
Turtle II has debuted its first technology in the form of a new A.I. chat bot Grok.
And now the technology will rival open AI's chat, and it's intended to answer, quote, almost anything,
along with having access to real time data from access, according to Elon Musk, who Binance.us tech editor Dan Howley has the details for us.
Dan? That's right. Elon Musk unveiled this x AI addition to the arms racetrack this past week, and he gave some hint as to what it has to offer.
And apparently it's supposed to be a kind of snarky or version of an AI chat bot than we've seen before.
He says that it's modeled after the Hitchhiker's Guide to the Galaxy, the books that are the guidebook from the Book of the same.
Look at where it's getting this knowledge. Oh my God. From unhinged tweets.
Okay, there's a war going on both sides. There's a massive disinformation campaign that both sides, both Israel and Hamas,
post crazy things, you know, from many years ago and completely lied to you.
So Twitter is like basically this this cesspool of crap.
Use real time data from X so it'll be up to date to that.
And regarding that, it zero. It's if nothing else.
Okay. Other chat bots, you know,
can just go in on Twitter and flood the whole thing with disinformation and this will go and grab them in real time and use it to answer questions.
Just imagine that. Okay. That's how they fanned the flames to spread more disinformation.
So that's all like weirdly crazy. So I'll leave that alone. The third thing that was announced was from up in the air.
And so they made the IGP do Turbo, which is like an even better version of Jeopardy for right?
It's all like, pretty neat.
But then I'm going to show you one new thing that's not even in the auto completion Djibouti store like Apple Store, Android store.
There's going to be another Jeopardy store. It means any of us that are experts on anything origami, communism, making sandcastles,
catching fly fishing or crawfish would, you know, worms, anything at all in the world.
Hang gliding. You can then make your own version of Jupiter and put it on the JPT store.
And somebody in the world can download that and pay them a little money, but basically get your expertise.
Imagine there's a pipe leaking in your in your sink, the kitchen pipes leaking.
Currently, you'll go on YouTube and find some crazy video about how to fix it. Right.
Or if you're lucky, like an E how or a wikihow step by step. But now you can actually chat with the plumber jpt.
And then it'll have all the knowledge that a plumber has.
You can even take a picture, actually upload that and say upload meaning straight to the Jeopardy and explain how to fix it,
because it can do multimodal things, do amazing recognition and know what is wrong,
and then actually start answering you and you can try it and say it doesn't work and you can give it some.
All right. It's pretty crazy cool. You can now create custom versions.
Look at that laundry body, huh? And then creative writing, tech advisor, maybe.
And resumé advisor. You know, maybe you and look job getting advice on how do get a job in Silicon Valley.
You know anybody that knows anything about anything can then, you know,
take all the expertise and then make special custom GP directions is like fine tuning the last layer.
It already knows like all of standard English, but then it doesn't know very specific like legal thing or some high end medical doctor thing,

you know, but now somebody can add that extra layer. Okay, It's a very neat thing.
By the way, you could do this. No, yourself people have done it. Okay. But you need to know Python programing.
You need to know how to basically, you know, train only the last layer, so to speak.
In a transformer. You're a bit technical, but now how do you make one of these?
You actually talk to the standards, repeat it. Usually you ask you put your question that answers you, right?
No, it's backwards. You tell it, STFU you and listen, I'm going to teach you something.
So learn everything that I'm saying to you and now your graduate. Let's freeze your and put it.
It put you on the store is need. Right. So watch this a little bit.
Beans of get that combine instructions, extra knowledge and any combination you can watch the text here Hey there it is I versus A.I.
And today, November six, Openai is rolling out custom versions of Techy Beat that you can see this.
You can make them yourself specific purpose. And there you start a conversation.
Deputies are like a new way for anyone to create a tailored version of techy beauty to be more helpful.
And you can teach your chess, can teach you trigonometric and teacher mission learning.
Whole thing is that you can have a deputy for one aspect.
Hey professors, obsolete work or another. Not so fast and not obsolete specific things that this is different mileage.
Maybe it's like you can just fill those out telling your activity how you wanted to act or how you want it to do a specific task or a number of tasks.
You can tell me all about yourself. So why would you need more of that?
Because like this Redditor so wisely said, you can create separate chats.
I mean, this look ridiculous, right? And kind of like about that kind of a price comparison websites I'm doing.
Price grabber is a bot that has a mine for when I'm doing work related to my YouTube channel.
For example, I found myself wishing that I had more than 3000 care travel advice.
So, you know, by the way, these things are, you know, actually even more powerful than this because the travel But the last one,
there are travel agencies that are still in the world. These businesses basically have no reason to exist.
Okay. But they exist. They're called travel agencies. People will go to them and say, I want to fly to Europe,
so make me a ten day vacation package that includes the following cities and they'll go and buy the plane ticket, hotel room, What?
What? Right. And give it to you. And they get some money. Obviously, that middleman,
that chatbot ecosystem can entirely do that because you can have a travel LGBT and then you can tell her what you want, Right.
And your boat, so to speak, or narrowed on the cities. But it doesn't stop there.
It can actually get the plane ticket for you. And how because you open a I made a plug in architecture.
They have a plugin API.
So if you then you learn the plug in API, then the API can call Saber, that is the airline reservation system and actually back the plane ticket.
Okay, I booked a hotel room. So just I mean, it's crazy.
Okay, what is all going to be possible? So most certainly learn all of this.
Okay. So learn to play the plug in API and do the attitude augmentation.
These are all functions in some kind augmentation. This is not achievable.
It's a little bit different. There are two ways to make charge a bit better than the average HPT.
One of them is called retrieval augmentation,
where you have an external data bank like the two homework examples I'm giving you and then you tell the chat to go and search it using a standard language lexical event or any database search Mongo or even Excel spreadsheet, run some kind of macro, you know, and get data and then use it to answer the question.
But that is outside the jeopardy itself. The second way to make the GP do better is to actually make the Jeopardy stuff better.
That's called finetuning. Okay, So two different ways that are trivial augmentation, call rag or fine tuning and or fine tuning.
So this is all about fine tuning, but it doesn't matter how you do it,
it's going to become like way better than just when this a single hallucination,
which is, you know, it can like just makes up words one at a time, but not a hallucination.

Actually, it doesn't need to be a thing anymore. They completely almost fixed it.

Tech Support Advisor You know, you call that your department while my computer doesn't come on.

Well, did you plug it in? No, I didn't know, you know.

Did you Pretty good tell you all that? Okay. Can be snarky if you want. Negotiator.

So, Chef Taco is Matt mentor.

I think in education there's going to be pretty amazing because many people that know a lot about things can make their own.

And then then so I would want to do it because you know, get paid for it.

So maybe I can monetize like what's in my brain by actually making the jeopardy and then giving it to the world.

Okay. So those three things I wanted to tell you that I need. Right?

Okay. So now we can do like our stuff. So we have, you know, these three bunch of slides and roughly about like an hour each.

And although there's only a half an hour left here, many of these things, they seem.

I'm not going to say they seem. Obsolete. They're not obsolete, but they seem a little quaint in the sense.

Almost all of these would be somehow, one way or the other, affected by the notion of larger language models, you know?

They are like revolution to transformers, to big architecture.

We need to focus on where this just magical thing called transformer is suddenly taking over everything.

So I'll go a little fast in the sense I want obsessed with every single little thing, but I'll tell you some neat things.

Maybe the only thing I can tell you here is actually two. One is a data structure called tree.

It comes from the word retrieval authority or IEEE, you know.

Well, so trees or data structure. So it's called a prefix data structure.

So that's where it's annoying functions reasons. The second thing is yeah, I'll tell you about something called edit.

This stands out of item. It's called an algorithm, a career change.

One word turn of the word hurry, change city, etc. to salt.

You know, you're saying minimum number of changes. I can change every single letter and I can change s to Q-on-Q.

Back to S. That's very dumb. So we don't want to do that.

You want to do it in the most efficient, you know, like parsimonious, like local greedy search kind of in a way.

And that is called Lowenstein algorithm. So I'll tell you about that. And we can play with it, but not necessarily obsess about the algorithm.

I think those are the two main ideation is, Okay, okay, So I'm going to start from scratch.

We did a little bit the other day, but I'll go a little quickly. The purpose of auto correction slash auto completion is very simple.

You start to search for something and then, you know, for example, may like this mayor off, you know, Los Angeles and purposely putting his own typos.

Right. So I said mayor of Los Angeles. Okay. And then it says showing results for mayor of Los Angeles.

It fixed two of my typos. And that's extremely useful because if we didn't fix the typos, it'll go in the inverted index and look for air.

And there's no mayor. There's actually a singer called John Mayer, by the way, but it's not Los Angeles.

So then my credit would come back with no results. Okay. Sadly, LinkedIn does that, meaning LinkedIn does not have a very good autocomplete at all.

You misspelled somebody's name, one character. It says a person is not found.

Okay, that is highly embarrassing. So LinkedIn needs to fix it. It's very stupid, actually.

Okay, but this is great. And said it found it.

So that's why the question is, you know, for example, all you can say mayor of Boston or something, right?

As soon as I say mayor off right. Look at all the things that people are search for.

So that is called auto completion, is trying to complete one of those.

You can pick it or you can type something or a correction, as you know, is misspelling summarization.

You know, I can tell musicians are no musician or something like musician like that.

And then it says musician says trying to fix it.

Now maybe I should do a musician like this, like that. So now it clearly fixed it because it is extra that it's called auto correction.

If you want both auto correction and auto complete together, they're very useful.

Again, when you text very quickly on your phone, that's actually what happens, right?
Like every time I type,
etc. because these next to it I end up doing data or if for some reason I'm off by one auto injector and it instantly fixes it, it's SRT.
And I'm like, Wow, thank you. Otherwise I'm spending I sending out typos. Right?
So that is what this whole lecture is about. It's very simple auto correction, auto completion.
So we can ignore some of these examples. You know, it shows you the same thing.
I mean, you can look at it if you type something incorrect, you know, that's a Russian mathematician in know submissive.
Right. And many people don't know how to spell them or out. Rehman Rehman is also a German French mathematician, so he's proud of his name is wrong,
but because are so famous people, it can go in the list of approved words,
meaning dictionary approved words, list of words that are correct, and then compare your misspelling with the correct spelling.
And so you meant that. So the way that is done is it takes the spelling that you type like submissively
typed wrong and suppose then it'll in this case it's completing but out of action,
all of the same thing. Ultimately, if you type something wrong, it will then take your word that incorrectly and do the edit distance algorithm.
I'm going to show you and say if I make one change, just literally only one change.
What new word, what I get and is the new word in the dictionary.
That means one change made a legal word.
You know, I made one typo, one little letter change, and then if I change it, well, I get a New York and then they'll make that to be the search word.
That's all your mental state After one letter change.
Suppose there's no legal word still. Then they'll try to.
Larry changes. Okay. And after I change it to letters, is there like a legal word in my dictionary And if there is or his answer.
So those how many changes you make, One or two or three is called edit distance.
Okay. So distance simply means number of edits, number of changes.
So in Google is really set at a distance of two or three, meaning the one try to make a very big typo, the one try to fix it for you.
Okay, then you on your own. Okay, that is all. So then that is how these things help you and wants to find the proper word.
Like in this case it's not SC, it's S-H. Then they'll go even further.
And so are people that type seven show correctly. The one in show business.
Tell them, Do you want that? So autocorrect first, followed by autocomplete because you cannot do autocomplete on the incorrect word.
So fix the spelling for the typo if there's a typo and if there's no typo, go straight to step two, which is autocomplete.
That's all. So there's lots of examples. We can skip all this, right?
So the whole underlying business owner, somebody some minor details.
I hope you don't mind. They just kind of look not really nice and many words, you know, and people cannot spell.
But the bad the real truth about all of this is throughout the world, you know, there are basically billion English speakers, right?
Most of them look just completely at spelling. Okay.
And then the newer generation, the TikTok generation, Instagram generation, you don't type anything, you don't write anything out.
So then you suck even more, you know, and then suddenly tech helps you.
It's almost like I don't care about spelling. The machine will fix it for you. So you outsourced your brain to the computer to actually what you did.
Okay. But some of us take pride in spelling properly. Spelling mistakes bug the crap out of me.
When I notice a typo, I go home and instead sometimes I pull up on the side of the road in my car and I fix it.
I'm like, It's embarrassing writing like a stain. Okay, but your mileage may vary.
Most people don't care. So that is all. I deselect that there are two ways to actually correct.

Okay. One of them is suppose to only type one word like inoculation.
Inoculation. By the way, there's only one. And that means get a shot. Okay.
In this case, you're going to get a flu shot. But in foreign countries, that's a good inoculation.
But it's not the word innocent. There's not too. And again, the only other thing that, you know, that starts with innocent is innocent.
Therefore, you think inoculation is also twins, but it's not. And that is what at a distance algorithm can fix.
Okay, okay, that's easy. One word. But suppose you type three words.
Okay, then they can do something different. They can take all the three words and search for what's called a tri gram, meaning Google.
Because of the extensive database that they have it. They invaded indexing.
They've gone through all the trillion words. Google actually has trillion words that they indexed among all the, you know, billions of web pages.
And the make up by gram by gram means any two words that occur together in any sentence, anywhere.
And how many times count, like, for example, here, say this, page one to Google.
Okay, Then they'll make a bi gram called the notes. They'll make a by gram called notes.
Are this like your homework notes are in in the form of and somebody else also might have said,
you know, this music is in the form of a chord progression.
So then the same the form off will occur, right? And they'll count the word, the font, that phrase, the form of one more time.
In other words, they can make a list of by grams, which are two letters, are two words.
Together are try grams, three words together for, you know,
I guess what programs and five words together six words and what rank how a massive list of all of them.
They already indexed those in the cache them somewhere.
So when you type something, then the search what you type with the tri gram by gram, all of that to see if what you type mostly resembles.
In other words, these two matches is almost matches. Then maybe that is what you meant.
In other words, we can use any grams they can use and grams to fix your spelling.
So two very different ways to fix your spelling once you started a standard algorithm.
And then the other one is to use down and gram database. Okay. Or you're in a combination of both, actually.
Okay. So then that is the bottom line that I'm going to tell you. And it's very easy, you know, just keep going.
One word mnemonic. I see many people don't know that P silent. Okay, so that's at a distance of 100.
By the way, I can start telling you what at a distance you mean. And it means it's all the same question from one word.
In this case, the incorrectly spelt word that is not in my inverse frequency index from one word.
How do I get to another word? It's all the same question. But except in this case, we don't know what the new word is.
But I'm willing to find out by making one change. Two changes, possibly three changes.
What do you mean by change? One kind of changes. There are only three kinds.
One kind of changes. Add a letter. Add a letter.
In this case, that is what happened. If I had a P, I'm going to get a new word called pneumonia.
Hey, that's an actual word. Bingo are the word add a letter.
Second is delete a letter. Sometimes you can delete a letter to add delete.
Okay. Which is the opposite of each other. I delete. Third one is modify a letter, change Z to ask.
For example one one letter change. Those are the only three and it's possible.
Add a new letter. That is not my previous word to get that.
In other words, a source where you want to convert that to a target word next that is going to be converted to salt again, for example.
So then the question is, you know, like what do you add? What do you remove?
And also what do you modify? Those are the only three things.
Again, every time you do one of those and then you're moving towards the other word that you want and you call it at a distance of 1 to 3,
you change this one at a distance. So in this case you don't need to change s at all.
You don't need to change at all. You need to change P to L Okay, So that's one change.
You need to change Y to T That is a second change. That's all you know, so that it can go from city to salt again, you're saying just to end it.

So those two words are supposed to have and at a distance of two, which.
I know the same words and source and target at a distance. Zero. No change to make.
Okay, So that's all. And that's why in here they're using the whole by gram index.
Okay. I call that the database of words. Well, the database of words meaning one word at a time.
That's a proper dictionary, but also a database of more than one word.
Word combinations, which we call Vikram's Trigger instead of Ram's.
Okay, so this is it. Spelling correction is extremely useful, as you can see in word processing.
Right. If you turn that on. It's amazing in Google Docs, you know. So Grammarly goes further.
It even tries to just get proper grammar. But as the start, at least like when I write in our recommendation letters to people, I quickly type them.
I don't look at it. Then I turn on the spellcheck. So important, you know.
So then it pulls up my name, it puts obituary, but I've added all of them to basically like an exceptions dictionary.
But then suddenly in graduate, I left out it just grow it.
Oh my God. And it says he wants to fix it. Okay. I'll think of fixing it.
I don't want to send a recommendation. Let's with a typo. And it looks pretty bad. So that's what it is.
Therefore, it's used in word processing. More certainly it is used in testing because, you know, the phone keyboard has looked so small.
Right. Look, fat fingers. So yeah, like after one letter. I could only type, etc.
I always type data out and hopefully it gives you all that possible proper words.
And I type once city fixed. It's fixing a typo. Free is great.
Okay. So therefore my my joke here. I hate autocomplete because you know it's a murder joke, right?
The word autocomplete itself is misspelled. It corrected it on the spot.
Rest in peace. You know, I would change this to P like a little green piece that you can eat.
Okay, That'll be even more funny. Restaurant and piece Restaurant.
Oh, that is something that is caught up your.
But okay. Fairly about you. Oh, yeah.
So. And then we became an email. Okay. Oh, yeah.
In other words, that is the wrong word. Read as a typo. But then that is not a proper completion.
Okay. I mean, it's a word, but then it doesn't go with the context.
That is where I comes in, so to speak.
So NLP Richard Djibouti would actually say enemy because, you know, it knows that that word doesn't belong there.
Okay. Anyway, so these are funny little things. I'm tired of your crap, huh?
I don't give a flying duck. Okay, Just look at that.
That's funny. All right. When you're such could be so funny.
Look at this one. Right. Katie k k, Kindle.
Kindle. Kindle. Direct Publishing, CDP somewhere.
Look for the word GDP. Okay, so Amazon has something called Kindle Direct Publishing.
So you as an author can write a book about something or a novel, you know, a textbook anything,
and then have some diagrams about possibly a plot that the Kindle can set a price on it.
$2. You know, somebody in the world can download a tablet or a Kindle reader, look on your phone and you will get money.
You're selling books. Okay. So anybody can be an author. They've had this for a long time.
But certainly now what I noticed since the last year was so much of the content that
got uploaded was generated by intellectual property and also stable diffusion.
You know, all this made journey image generators. And that's pretty bad.
So Amazon had to revise their terms and conditions and said if it was entirely air generated, you didn't modify it.
You have to declare when you upload your book and say parts of this are air generated.
On the other hand, if you then took something that was a generator charge would be to give it to you.
And they changed some of the words that that is called air assisted.
In both cases you have to declare it. So then the user that downloads your book knows that you didn't write it all from your head, okay?
That's why they're trying to solve this problem as very fascinating. So you need to basically, you know, do that.
And if you're found to violate the terms and conditions elbon, you take all your money back, probably never sell books on

Kindle again.
But if you go to Amazon right now and look for travel guides, it's pretty bad.
There are so many little guys that are entirely fake that it may charge a pretty little say.
The Traveler's Guide to Greece. The Traveler's Guide to North India.
However, Grant And then there might be photographs of places that don't exist in North India.
So if you actually plan to go there, there's no place there, right? I mean, it's so stupid.
It's not on the map because it is making up words. Making up images is actually very bad.
So Amazon has a pretty big problem against polluting the, you know, content.
That's why they had to come up with these terms and conditions. It's fascinating.
On the one hand, technology enables. On the other hand, technology also helps the bad guys do horrible things.
Look at this one. So once again, I mean, this we can read afterwards.
You know, you can go and read it. But I wanted to show you, though, this was written about a month ago.
Very, very interesting. So online search, you know, again, what happens in our the search.
Yeah, you are, you know, sometimes get good answers. Sometimes you actually get pretty bad answers.
So borrowed and on like open air jeopardy is able to do a Google search and that is advantage.
Okay. So I'd say distillers Tibet because there's all kind I mean that in a paragraph
of mostly factual sentences suddenly this one sentence is completely wrong.
You know, I think I told you the example last time telling me where the names of 50 US states came from,
and many state names are correct, the history. But then California's history is completely wrong.
It talked about, you know, Christopher Columbus that had nothing to do with California.
So then that's all like really bad. So this article tells you again, you know, how are they able to fix it?
You know, the honest answer? I can tell you the answer at some level, Right?
They cannot fix it. It'll never be 100% perfect. Just cannot.
I just tell you that's what I believe. So then that's a problem, right?
In the future, all this is great. It's a accepted technology. But there still can be something so bad.
Anybody killed somebody and a brand new lawsuit, you know. Then we're back to square one.
I think it's an unsolvable problem. Technically just cannot do it because there's no meaning inherently in just only a bunch of
words or only a bunch of video or audio or anything really knowledge just in the world.
So if you don't live in the world, you cannot possibly know everything, but you can read about this.
Okay, so back here, spelling errors, this all just, you know, some statistics, okay?
It just says, you know, like how many like what percent of queries actually have like a spelling error that these are quite big percentages.
Right. So let me say this. A word on the search engine side to have things like autocomplete, autocorrect.
So it's a justification for why they need it, right? Like here.
Otherwise the search engine will come back and say search results empty like LinkedIn at all.
You type your own name with one. One word of it is a person not found.
You know, that's very embarrassing. They should actually find that person. Okay. But yeah, then that makes the system look harder to use.
It's painful. All right. So again, detection correction, right, is pretty easy.
So any spell when you find that there's something wrong by looking at the word in the dictionary,
I type something wrong with the musician and they go in search and the index with my incorrect spelling, you don't find any documents.
So that's what tells you out. Maybe there's something wrong in the query word.
Then you try to do at a distance or engram and say try to fix it, and then you can, you know, serve the user.
So that's really all. Okay. Say when you have an error detection, right?
Like with the autocorrect knowledge, my worst enema, what happens is so supposed to say autocorrect is my worst and I'm supposed to say that again?
So I put it up. Autocorrect is my worst enemy. Clearly there are two different things you can do and enema.
Or you can say enemy, right? But which one would you pick the?
So now suddenly you need the context. You actually need to know natural language understanding in a medical in a doctor's office.
Then you might say, the nurse that might be a nurse, might the doctor might type you.

The nurse is going to come and give you an enema, but maybe the doctor type an M.D. and in that case, the text should actually correct you.
And I'm very sorry answer in this case, you know, is your worst enemy.
You know, then I type in MP, then it needs to actually say, Sorry, this is enema, this enemy.
Okay, So depending on what the context is, the system should either fix it this way or fix it this way without context mathematically.
And I'll just have to pick one. And then if you pick the wrong one, then that's when it becomes a joke.
Become a meme. Okay, So that is a hard problem, basically.
All right. So then regardless, you know which to fix, they're always trying to fix it because it's easy.
You know, if you have two words, just pick any one word and show it to the user.
If that is not the right word, then there's still no there is something wrong.
They can go fix it themselves. Okay. So it's always trying to correct you if you make a mistake.
So then here are the two ways in which you're. Good point.
It actually will. So one way of getting the context is look for like more works.
Exactly. So, again, you know, the nurse is going to come and give you a you know, an empty.
Absolutely. In a nurse and any. Those words go close together then nurse and enemy, you know.
Yeah, totally. In a battlefield, supposedly in EMT is just around the corner.
That's absolutely enemy. Yeah, sure. Right. Good.
Okay. So. Yeah, exactly right. That right you can do. And anagram matching or at a distance.
Okay, so let's one. Right. So stuff like that. Right.
Suppose you say grass then. That's most likely grass.
Someone that left out a letter, I, let's say at a distance of one because I do one instead.
Okay. That's pretty easy. This one is where the problem is.
If you just say like, three, then three itself is legal.
But maybe the grandmother and then grandma tell you they're like, Suppose you say nobody lives there, so put it up.
Nobody lives three. Then worried about where it is, correct?
Nobody lives three. But then the whole phrase is meaningless.
What do you mean nobody lives three. Then it should correct the 3 to 0, even though three and there are both correct words.
Otherwise, there's no grammar, there's no spelling mistake. But you still need to possibly fix the word right.
And grammar can help you get like twice, you know, stuff like that.
Many of us, you know, in a hurry want to say, I wish you peace? Maybe.
Maybe, you know, the SWAT system or the spellcheck. Maybe.
So if I start typing PR something and then they put this up and you accept it, so that's also pretty bad.
I was sure bef okay. Then again, same thing. You have to change it to peace.
And these are problem because they're both correct. By the way, so many people in the US and I sadly, you know.
Oh, I don't know. Ah yeah, you're crazy to type your crazy.
In other words that type your crazy. It is so ugly. Please don't do it.
It's disgusting. It's gross. Completely wrong. All right.
Now, of course, I tell you. Okay, Well, I should tell you.
Suppose you go to Hawaii, you might find a sign on like this.
Nice little, you know, this store right by the beach? It might say.
That. Oh. Cowabunga. And then you have this right next to it.
A different sign that says. I don't know. I'm going to sell DVDs or go into Let's pick something.
I'm going to say DVDs for sale.
You. You. That should go over here.
Surf's up. Contraction, right? Surf is up. For God's sake, when you do a plural of something, please don't put an apostrophe there.
It's so gross. So many people do it, though. Wow.
Okay. So that's what this is. I guess the point is these are legal.
The word there's nothing wrong with this word is not typo. But you might need to fix it to this because that is the proper usage.
Okay. Great. And otherwise, not a quite easy problem. All right.

So non word spelling this. These are easy to fix. Okay, Because you actually made some typo to the word the grammar for a fee.

That's not in the dictionary. They can easily find that because they go looking for it in the index tfidf.

There's not a single document. Wow. There's a typo. Let's fix it. Great.

So then you need to go searching the dictionary.

In other words, start making these and grammar data at a distance changes and then search in a dictionary for dictionaries.

Very small. Then even after you come up with a legal word, it might not be in the dictionary from I think what is still a typo.

But if you're dictionary is large enough, you might be able to catch it. That's all that says.

Yeah. Then yeah, exactly. Same thing. Clause means smaller at a distance.

Can I make one change? A plus or minus or a delta somewhere and then get a legal word?

No, not not. I didn't find any can I do to change this? Three changes.

Alright. I'm not sure. You said a distance. Okay, that's all. So given any two words like you in the word dictionary and the word spelling,

how far apart are they that commonly letters should I have to change like ah, delete modify to go from A to B?

You can go from anything. Anything. Again, if you have very dissimilar words there, the distance is going to be pretty large,

but otherwise, you know, you're playing these games, okay, They're coming like the back of magazines.

How do you change something to something? Yeah, it's fun.

And I show you like a little JavaScript interface and you can type in the two words will actually show you it's very little table,

and the table is the whole one word. First word, second word source word.

Target word. I like a matrix too. Good. I'll go on filling word change to make a change, not to make, and then keep on adding the change.

Everytime you make a change. The Terminator number one, two, three, four. On the right hand side it'll be the actual edit distance.

So purposely I won't bore you with the algorithm I want.

Interestingly, the algorithm you don't need to know for the exam either, but then at some point it's like a triple loop looking look at it.

It's a nice recursive algorithm, but then we won't spend time on it.

But we'll just know it's called the error distance algorithm. You can Google it if you want.

Okay. So we use a basic, you know, kind of an idea called a noisy channel.

So what we say is when somebody type something incorrectly, it's almost like the brain got a little foggy, maybe the mental type, the right word,

but the right word got passed through this noisy, you know,

this thing that corrupted some some letters and then the corrupted word comes out and it's a typo.

And then the question is, how do you denounce it? The card you go from type of back to the real world.

So we call it the noisy channel idea, philosophical thing, almost noisy channel.

This also, by the way, is so similar to a stable diffusion. So in other way, stable diffusion to our generator works is it's absolutely amazing.

I don't know. It shouldn't even be possible, but it actually works. You're taking image, you're adding noise to it, Screw up some of the pixels.

Okay, that's training data. Perfect image, slightly screwed up image.

Then take the slightly screwed up image and even more noise and screw it up even more.

That's training data. Do it one more time, take that more screwed up image.

Ah, more noise, more training data three years ago. Keep on going at the very end you have perfect noise.

All the pixels got changed completely. Noise. Okay, that's training data.

Going to take all those rows, individual image pairs that went from a good image to complete noise.

Maybe through a thousand iterations or million iteration. Train in your network to say, learn the pattern.

How do you go from this to this? How to go from left to right, left, right. Then you could do the most meticulous thing.

You can do it backwards, completely. That generation, I'll give you a pure noise.

Turned it into a cat. That is exactly what I told the stable, deficient algorithm.

Okay, It's a little like that. The whole noisy channel idea. So once noise has been added, how do you remove the noise?
You know, I go off on this tangents and I hope you don't mind, but they're quite useful tangents.
I just want to show it to you. It's called stable diffusion because that's exactly making a typo.
And then we'll get back to the Promise algorithm because you can actually see what I mean, the whole diffusion idea.
So those are diffusion pairs that I'm talking about. I'll just find one example, possibly even this one, you know,
So you go from one image to a slightly noisy image and they say that's a training pair that makes two
more noise is another training pair that the more noise of the training pair that the more noise,
the more nice to more nice. At this point, that image became pure noise.
But they're all training in pairs, like one pair, second pair, third part perfect.
But in reality you'll have many thousands of pairs. Okay. And then you can learn how to then reconstruct.
That takes the noise out, which means you are able to do image generation, which is completely crazy.
Yeah. So please learn about stable diffusion. It's great. Just look amazing over time.
Plus additionally also trained to minimize the loss in the text, meaning already describe the text.
You can combine the contrast of loss. It means then you can type something like make locations in an amazingly it'll just be created.
Not that image, but an image like that. Okay, so that is the whole idea of stable diffusion.
That's the noisy. Now there's many types of many type, like some works hardest in make.
That in the art it is actually the notion that some high level conceptual level.
All right. So hardest misspelling happened this really easy little slide.
You know, you type very quickly and maybe a key bounce is set to high and then you don't notice it'll go one next, become two axes, right?
And then hardly the corner is a correction. Or sometimes you know this what I do all the time.
You're off by just one letter when you type really fast. So rather types of important.
Like if you look in the keyboard next to each other in the bottom row. So important anti refresh slide to my right.
It's nighttime. I type empty and I can fix all that. Right. Likewise.
You know okay so English is pretty crazy as you know right? Normally in many, many, many words e comes first and then followed by.
But that is not universally true.
Like, you know, like you have received, for example, is IEEE or Concierge, the person that's in the front of the hotel IEEE.
Okay, then that completely throws you off. If you did not know that, you will always do it before I am an email typo.
All right. And then likewise, you know this a slash killer.
Microsoft came up with them, but it sucks, right? One way. It's called Silverlight.
Okay, So Silverlight is one word that I put in here is you put like an extra space because you did not know it was one word.
So typos can come because of so many reasons. These are all reasons or Kinect.
You know, the little algebraic depth measuring device that again, Microsoft made.
So that's a new word that the made up there's no word called Kinect. Okay. This kinetic energy, but no word called Kinect.
So somebody doesn't know that. But they have a friend called kin here. And this is spelled with twins.
So I'll spell connected twins. Okay. So they can fix that again, because this should hopefully be in the dictionary.
Okay, so many reasons for misspelling, but they should all be fixed. Right?
This one is again, Google actually has all these crazy, you know, call lists that they make of famous people,
in this case, Britney Spears, because so many people did the search for Britney Spears.
Right. It can be spelled in like 500 different ways.
Oh, my God. You can spell Britney Spears this name in 593 different ways.
The devil is always. You can look at it. I mean, it's nuts.
And likewise, you know, so many people spell her name correct. It's not that hard.
Okay. If you look at it one time. But the problem is people don't look at anything.
They simply make stuff up when Britney, you know, and also unfortunately,
in the in the US parents come up with very creative names to name their kids then they are not get lost in the roster okay own be different
so every name is spelled in every possible way you know possible and so then you have no idea where there's a ground

truth anymore.
There's no consistency. Okay, So then somebody named your friend might be called Britney.
What the. Anyway, therefore, what are you going to date Britney? Or if you are Britney, what are you going to do right now?
Okay. So the thankfully, the good news is there's been so,
so many times that that becomes part of the vocabulary diagram and then Google fixes it immediately for you.
So, again, you can be lazy, you can be ignorant, they can fix it. What about this context?
You need context. Okay. Okay. Suppose you are flying from Heathrow to L.A. again.
You fly from place to place. B Right. So the form more certainly would need to be changed to front you in the form is a legally proper, valid word.
There's no a typo there, but you still have to fix it. Like, Oh, this is way crazy.
Okay, so positive policy already. And then they come back in 5 seconds in the video.
See, already they need to fix them into two different words. This power cord sheet that this a power cord.
No power cord so courtesy already the group again whereas video card is something you plug in your laptop GPIO card right That's here They can't that's not see already it has to not so the word video or the power word,
we're disambiguate it to tell you what CRT should change into and what what we're saying is support your CRT.
That is distance is one. Whether you insert CRT or whether you insert CRT, they both are at a distance.
One, they're both legally valid words, right? But actually you pick your topic based on the word amazing.
So again, the program is going to help you because in Bagram, that's the Bagram.
That's also Bagram. So this is close to this. So they'll fix that to that.
Likewise, this is similar to that set down. Okay, cool.
And grams over a number and grams for the win sometimes.
Okay. This is the whole phonetic spelling. Okay.
So many people don't know how to spell many words, so they start as if they would hear it like diarrhea.
Okay, let's put some crazy work in. Diarrhea is a double r h.
There's no way. And heck, you would know that the word diarrhea as h because you don't spell the letter H is silent.
So then that is a phonetically similar word. Okay.
Okay. So based on the pronunciation, you misspell your word. There's an amazing algorithm that I'm not showing you in the slides.
It is called sound X, okay? And the sound decks is meant exactly correct.
Those kinds of errors. In other words, what are some common words that people misspelled because they spell it like they would say it?
And what is the proper spelling? It's like lookup database.
Okay, so then look in the sound dictionary to see if your misspelled word is there, then don't even do that whole background business.
You know the another way to fix it. So there are many ways to fix spelling errors.
Okay, that's that's where we're going with this. All right.
So then again, you know, I mean, stuff like that, if somebody.
This idea like, Hey, check out this idea. Then they left out the spelling and then that word is not in the dictionary, clearly.
Okay. But then they could hopefully know that there's going to be a space in between.
Likewise, the proper way to say in LA is with a hyphen. But many people don't know that.
They just type in LA for these kinds of cases, by the way.
Google actually has separate dictionaries because of the common misspellings and they'll look it up
in the dictionary and all the dictionaries are sorted alphabetically so they can do a binary search.
Okay. So this one in our chat in Spanish. So two problems here.
A space is missing and Spanish is spelled as Spanish.
Spanish. You know, I feel like I don't know French or somebody. Right.
S-H is actually pronounced THC or even are it Spanish itself to reach it.
Then, you know, people actually would not say, say statistic expenditure and you've got to fix it.
So two errors there, right? Okay. PowerPoint is not two words.
Microsoft made a crazy product called it PowerPoint. It's one word. So if you did not know that you will say PowerPoint, then they can fix it.
So it's fascinating, right? This one. And you know, this other chip company comparing the bottom, it disappears.

Right? So then that's a new word that is not in the dictionary.
Right. So who knows what ambitions for. Oh, and actually an acronym.
Look it up. Okay, then. In there.
So positive. Empty processors. All right. You need to actually know that it's not a typo.
So those who are and really are, even though it does know what Nvidia, Nvidia GPO, they should not flag any media as a typo.
So those words can be added to the dictionary words when many people search for something.
You can make that part of a dictionary even in your home in Google Docs.
You can add your own name as the exception. When you see my name, don't call it.
I know a typo. When you see Henry Salvatore, you know, computer center, don't call Salvatore a typo.
When he when you say return be, don't call. Which would be a typo. Just add it to my lexicon or guide in my dictionary.
Pretty simple. By the way, a fascinating fact took Nvidia and.
Okay. In fact, let me see. Who knows, just for fun. What is common to Nvidia an M.D.?
This an obvious answer. Just tell me. Yeah, they're both chip companies.
We know that. Okay. What else is common to both of them?
Anybody know? The CEOs of both of them.
They're cousins of each other. Oh, my God. Amazing.
Look at. Whoa! Her.
She's a CEO of R&D. She's actually Jensen's cousin.
Oh, my God. One family rules the world. Okay.
Those two companies are amazing. I mean, Intel is separate, but it is actually funny.
Fascinating. Yep. They're rivals in some sense, because AMD also has chips built into it.
Sam bought this chip company called Radian. San Media had a competition called Radio Ready and Major News.
Sam They basically swallowed Radian for like $4 billion. Okay, now, say AMD's chip was built into it.
So then they actually go up against and radio. So I guess it's friendly competition, right?
Snip. All these things, I tell you.
One. Yes. So in other words. Okay, the noisy channel business.
Okay. So some word goes in. Your brain is foggy. You made a typo, right?
The resulting bad, the incorrect word. We call it noisy channel.
It's a misspelled word. And the idea is, how are you going to remove the noise?
That's like encoding noise and then decode noise and get back to hopefully that the proper word again sometimes is ambiguity, right?
You know, based on context, a power card, which is really you got to pick the right word.
So conceptually, that's all we think of typo as some noise card. Ed By the way, noise can get added to bits when you transmit them.
Hey, here's here's a very simple question. Oh, okay.
So what is a very, very simple algorithm when it transmits 11 bits of something and then you can assume that only one bit is correct,
that okay, no more than one bit of scripted. What is algorithm to fix it?
A single bit corruption. Pretty nice.
Oh, my God. Oh. His work.
But yeah. Something like a checksum, correct?
Sure. But what is algorithm or what's the name of the algorithm? Right.
One bit. That's a pretty big assumption because you know, more than one can.
In fact, it's an even bigger assumption.
Suppose I transmit 15 bits, okay, for about 15 minutes, then 15 bits I would pick off like maybe the second bit and the fourth bit date,
but write one more as what I call error correction bits.
Okay. So I'll stuff my 11 bits along with four other bits.
So I have 15 more bits. Those four bits we assume 11, but we're not encrypted.
Okay. But in the remaining 11, only one gets corrupted. And the idea is to transmit all the 15 to the other person.
The receiver. The receiver would then grab this and keep it aside because we know that those are not
corrupted and then do its own computation use from that 11 and compare with this.

And if there's something different, they're not the same. That's when, you know, something got corrupted.
And the second amazing thing we can do is we can know exactly how that 11 bits what is the position where it got corrupted?
Say the position was here and you can flip it. Say after corruption you found out that that British corrupted say says one now to continue to zero.
Conversely, if after corruption you find a zero, you can change it to one.
Wow. You should look it up. Okay. I'm not going to tell you. Okay. I'll ask you a moral question and a good one.
Okay. That is a pretty big assumption that your error correction bits are themselves not corrupted and then are relevant, which only one bit can be corrected. Not to not. And it can fix it.
Find it and fix it.
What about a more complex algorithm that can do much, much more like almost, you know, like a mark of chain kind of analysis and then fix it.
What algorithm would that be? Or this embarrassing last part when I tell you the RNC will reject the victory algorithm.
Oh, my God.
Why do you think, Andrew, to be give us all this money in our history thesis at USC and the State Department is the better be scoring algorithm.
It's an amazing algorithm. Please look it up. So that is the reason why he became a billionaire.
And then he was able to bring so much money back to us and named the whole engineering department after him, engineering school after him.
See it? Is that okay? So, yeah, maximum likelihood.
It is so incredible and it can basically fix your cell phone transmission, satellite transmission and any traction in the world.
And what about that silicon? That is bad bits.
Good bits. Players, please learn about this. It it does not get better than this.
Every cell phone company in the world, Apple, Samsung in our Android, doesn't matter.
They all have the license algorithm from Qualcomm. So he and Qualcomm became instant billionaires.
Right. All right. So anyway, so that's all really nice stuff. Lowenstein Then maybe a little break.
Okay. So Lowenstein is very simple.
What if you are able to take one misspelt word and make either an insertion like a plus sign, right, to add one letter character What, one character?
And then you delete again or you change the fourth one as well.
But that is not in the Loewenstein algorithm. But then Google would have to take care of it, which is it would transport letters back and.
Okay. TRANSPOSE Well, you know, in a way transpose also. Lowenstein can fix, okay, because you know, it's some edits.
You have to make pretty good principles that you are transposing. So I'll show you some examples.
Yeah. So this is so cool. Thankfully, many people make errors that can be fixed after exactly one night a distance.
80% of the errors can be fixed with one edit distance and then within two edit distances almost all errors can be fixed.
That is why the search engine will give up after two edits.
If you still don't find the change, any change that you make in the dictionary, you'll say sorry.
No look, not no results. Then I have to say, Wow, I made a typo. Okay, so it didn't do it.
I can fix it. Okay. All right. And then likewise, you can also then take the the one word that you fixed and then for context, go in the engram and then in OSI supports it, fixes multiple possibilities.
Okay. And programs give you this, it give you CRT, NC already but that was video and check.
Is it video or CRT or video? CRT video, CRT.
So that's where the engram comes in. So they're both, you know, greater distance followed by engram.
Okay, cool. So probably means again for video card, maybe there's some crazy thing called video cord supports of an HDMI cable.
Okay, Somebody might call the HDMI cable video CRT or CRT.
That is a lower probability. Okay. That's right.
Google will tell you they have an anagram table where video shared occurs many, many, many more times is a higher probability.
So go with that. Let go and pick a CRT. Cool.

Right. Okay. So these are very obvious common sense but wonderful things.
Okay, so then that is pretty much it, right? Okay. Lowenstein I'm going to show you.
Lowenstein So check each grade.
I mean, I suppose you've got a phrase typically video card event to check every single word for any word that is not sound.
Again, they made a typo or somebody made a typo so it didn't distend one or two brain.
See, there's a proper legwork then going to engram index.
So that's the modification. In other words, a pure spelling correction algorithm is only just.
Lowenstein Okay, but now they can add the engram stuff to learn standard and so on.
Better. That's it. Again, like I said, you know, some words are so commonly misspelled that you can catch them and as soon as it's a misspelled word,
that's probably the first thing that they do. Check to see if you misspelled way is already in a table with look up with the proper word.
In fact, you can hash the respect word, okay, you can binary search for it.
Suppose you don't find it here. Then you can do this part.
Do the Lowenstein followed by by context marketplace and right this a lookup already in the lookup server always do the calculation.
These are fun. Okay, look at this one with an edit distance of one.
So push it up across AC. I'm going to show it to you here.
So step backwards. Incredibly that one misspelled word.
I can do just one change anywhere and I'm going to make six new words that are all valid.
So then in terms of context, is it like a valid word to fix?
Right? Only an anagram can save you in Crash.
I can do an I can do a plus. I can do a insert to here ac t aureus actress neat.
Likewise in acrostic and delete the letter A and I have crest crest or you can do crest that is a transposition
but change the letter Siena crest and then I can do change the letter R to see that substitution.
Then I have access access to a course across the letter.
Another change I can change the letter E instead.
Here I change the order to C okay, but I can change E to all and I get across, you know, so many crazy things I can do right now.
Other words, any of these could be misspelled to be that. And finally, last but not least, acres and acres.
Take out the letter S. Wow. So many. They're all one one letter change.
Mr. Lo, So many actual legal words in the dictionary. So which would it pick?
N l u you know, or n grams. Ten grams is a pretty cheap way of natural language understanding, by the way.
But for a better natural language, understanding things like tangibility elements.
Actually, actually what you need. Okay, great. Okay, This is fun.
Okay, then of all of those, which one you pick, right.
One answer might be Usain Ingram, for example, who was the actress who starred in, you know, Star Wars part one.
So then you type this host, the actress who starred it is most probably actors, right?
These words don't even apply. So then you can say, who is the actress?
You know, that's a big and grand foreground to fix them. Or you can say in all the web pages, in the billions of pages that are ever indexed,
I have a frequency for how many times I indexed this, How many times? How many times?
That's a probability, right? So go with the highest quality. In other words, take the most commonly used word game.
I mean, many of the people in the world won't say acres of the word. You want to say acres or how many acres of land.
You should buy it for me, you know. Ah, Chris, I mean, that's a common word.
But when was the last time I said, Hey, I want to hold you and Chris? You mean anything to do?
Right. But, you know. But a much, much more common thing people search for is actresses.
Then go at that, right? That's all. So that actually and I like to approach I like a cautious thankfully a much more common you know across I'm across from you know or the cities or customers or places across.
So go just by usage see this real world and convert to frequency.
You can take this and divide by how many documents you have. And then that's where the probability comes from.
Okay, that's a big number. Or you can divide by the total number of words in the dictionary.

Some some common division through all of this. Okay. But these these are all séquences.
So in my index across occurred so many times, one 20,000 times compared to 12,000 times.
Therefore, I would not change a into acres. I would rather change it across.
I have a higher chance of being right. Pick the highest probability. Maximum likelihood.
Okay. Just another name for Harris? Probably.
Okay, great. And then. Yeah, then search for correction is left.
Right. Meaning to take a word in English right from left to right.
Correct to standard word spelling. So then you try and fix every single letter, try and insert delete, and all are from left to right.
And as soon as you find a match, stop, because it answers the question for any misspelled word.
If I do like all kinds of substitution possible in all over the part, if I leave out and then I can even.
Huh. That's pretty funny. I can change it and I can make a big word called for climate, for help.
For climate. Oh, my God, That's so fragrant. That's also a word.
But then I wouldn't do that.
I would get the E and get the enemy on animal before climate, because it can go like one letter at a time and find that right.
So that's what this means. I can't.
It just simply means broken, screwed up, damaged after a Swedish day forgot to look for it.
Okay, so EMT can become comforting. I don't even know how to spell.
Actually, I don't. The. My God, I can't.
Whoa. Oh. Oh, my God.
What I saw was German volk.
What I saw. I didn't do this on purpose, by the way. I swear. Or so.
Oh, my God. Okay. For a clown. Oh, I'm so sorry.
Oh, my God. Hashtag fail.
I pride myself on. Good. Okay, but. But not in this case. All right.
Yes. Yes. So the whole notion of prefix matching.
So I'm going to show you this data section called Political Tree. We'll actually take a break now.
Perfect is very simple. You take all the letters and, you know, in English or some other language.
And for every word, an actual word, it it'll be a leaf somewhere.
So how do you get from the letter B to a blue buttock, obviously.
In other words, bluebirds starts with the beep. Bunting also said with the beat. So anything that starts with C will eventually become mass.
A bunch of leaves. All the letters I said would see in the dictionary would be leaves.
Okay. Likewise, anybody that starts with D would be like a big bunch of energy leaves.
But what happens in between the leaf and the word the you would sort them alphabetically.
In other words, you would have all after D you would have all the words starting with the letter day and after day,
all the letters starting possibly what day it There's not a day that would be double that kind of a
data structure where you organize every single word based on the first letter and then the same,
then two letters being the same and third letter for life after It's called a trade data structure trait like this.
Right. Like this trade data structure.
And here we call it a prefix tree, because you talk about all the letters that are going from like left, right, prefix as opposed to suffix tree.
Okay, that's like the opposite. Okay. So there's all kinds examples.
You can look for it here. Okay. Like, for example, the word and the word end is a followed by n followed by end.
But the same and can also be a prefix for entity, you see.
That's right. And then are common to end and end. So you can take every word in English and they'll all become really leaves here.
But meanwhile, between the leaf and root, the road is just simply all alphabet.
Okay. What is in between? It's all organized by prefix.
In other words, after a there's even more, by the way.
There's ab, ac 80. All that is not strong. It is just shown for what is an OC saying the anti.
There might even be more you know that connotation.

There might be an end somewhere and you can keep going further down the word annotation or be like way further down.
But they still start in other words, annotation shares with and an end.
The following thing the first letter for two letters they all start with an okay.
Yeah. And so there's many examples of prefix trees when you go through all of them.
C t and TED, for example. Right. And then in and then in.
And over here is instantaneous or maybe integer as the laws start with the letter.
And it's a great idea. It's like a tree, isn't it? So we use a tree to actually do the matching.
Meaning we started the first letter, then go on the tree and eastern to try and fix what is in the tree.
If you can get like a new legal word or not, great.
Okay, so. There's also like a Celica, a suffix tree, right?
There's a perfect tree. There's all kinds of trees in the world. So something straight.
That's pretty much the opposite. Okay, so in such extreme, let's look backwards in meaning from the matches, from right to left.
All right. But that's not important here. I'm going to leave out the complexity.
No need. Okay. This is. We can also do one more.
I told you. Common mis, common errors. They've already been entered into some kind of a look up.
If this incorrect word occurs, I can tell you which actress.
There are many of those collections of them. So Wikipedia has one of them lists of common misspellings.
You can say it. Then people like Google, they have all that already indexed, meaning that they have the table already in the search engine.
So they will then look in that table. Common, common misspellings.
To see if you can very quickly identify the misspelled word selector.
See all of this controversy. Controversy.
That's the misspelling on the likewise duck connection.
Good luck spelling that said to right. Many people think it is right but it's that and so on and get rid of them.
So therefore you can easily look through and look at one of these and fix it before you do any trade, before you do any.
Lowenstein Right. And sophistication. So where else can you look like Who else made a list like that?
This one is spelled out net and then requesting, you know, but, but we don't need to go through all of them.
But you can go to it. If you then make a table with all of them, then you can quickly look at look at it,
look for the misspelled word and very quickly return all the proper expelled words.
The ACARS would then give you all the six words actress across acres, Chris, you know.
Look it up. So look up the easiest. All we can do. So then you can correct like just basically typos, you know.
619. Maybe I'll go to 620. Okay. Send them.
Know exactly what we say it is. You're done your homework then, Grams. And that's a very cool and grams in a way capture the language, you know,
because this you know what half million words in English are we commonly used?
Those words are not all paired together or triple quadruple together in any random order if say the same things in many of the same ways.
Hi. How's it going? You know, after how is it most likely the way it is going?
Nobody says to you, how is it happening or how is it, you know, tomorrow or something.
If somebody says your how is it mostly how is it going? Therefore, the English usage is captured and all of this in grams.
And then we can then, depending on how many words the user type, if the user types three words,
and then the three words are not already in the term frequency dictionary,
then I look at the programs with those words and see if they made a typo or not.
If they are typed forwards, then I'm going to look for any four four grams white program spectrum.
So for example, so as the independent, you know, independent, the made to type is independent.
What is turned into s, if I look in the foreground, I'm going to find it.
Hey is wow. Likewise.
By the way if you look first serve as the starting point and even the letters and the prefix tree will tell you I m this comment.
All of these meaning all of these are matches. Okay. But then given this, I'm going to go with that because that is the highest frequency.

And I also see that. Okay.
And then there the distance is two because I can change this to I can change that the but we don't need to do elementary,
then we can just simply do it for grammar analysis.
Okay, let's do a break. Let's do a break till 626. Right.
A five minute break. Yeah.
More to tell you more. So John Tallman, who was just here talking to me.
You know, it's so cool, you guys, how you're able to take all these things in so many different directions.
Okay. So one thing that you have to do is when you take an El Alam, you do the tibial augmentation.
You know, there's two pieces of style and piece and then documentation.
There's the extended database piece. Neither one of those in a real company,
they'll allow you to have it on a public cloud somewhere because it is your protected intellectual property data.
There's no way. Now, [INAUDIBLE], DreamWorks, for example, there's no way the DreamWorks movie script would ever be on the airplane.
So it's nowhere at all. So we need to then have both of those.
Both of those local is not a problem, but it is, you know, you can do it Al-Alam, it can be local, meaning you don't have to pay Chad Djibouti money.
There are so many. So I have a separate lecture I want to mention and get back to this.
But pretty quickly there's things like Mistral. It's an amazing Al-Alam call.
In other words, Djibouti is exactly only one of many, many, many possible larger language models.
There are literally dozens and dozens and dozens of them. Amazing.
And this be one of the biggest really, because it has 1 trillion parameters.
It's massive and it belongs to open air and not to deliver it. But if you had a cool alternative where you can then have some of them not that big,
but running on your laptop or running on your local server, that's amazing.
Okay, that solves one of the problems. I have a local alarm. Next is you want to have a local ritual augmentation.
Meaning I want a local knowledge graph. I want a local vector database.
One pdf file I can chart. It's fine. But what if I have 10,000 PDF files?
Okay. Then suddenly, while I'm running out of despair or something. So it might be cool if you have a private network of some kind, you know,
maybe in a peer to peer network and then store your external drug stuff right
in the network and have your Al-Alam access the data from any of those nodes.
Bitcoin, not Bitcoin really blockchain. So blockchain actually could be a solution for this because in the blockchain, somebody that's part of some private blockchain, they all run the same software, right?
And each block needs to be vetted by everybody. So classic blockchain problem.
So then all the blocks that you add might actually be this kind of actual augmented, you know, pedophile blocks.
So then it sounds like so many problems, right?
And encrypted and you can verify that it's authentic information or it's private and it's not on your machine alone.
So then you can collaterally share some kind of a distributed database, but it's not a public cloud.
Okay. So it's a great idea. So I was just talking about that. So anyway, if you want to know the alternatives, there are many of them.
But then here's one of them called Mistral. So Mistral is incredible.
It came out of just France, you know, from Norway. Okay.
So Mistral is only 7 billion parameters and it can give you like 94 or whatever percentage of energy pretty far.
There all the metrics, and I'm not going to go through it. Okay. But it's so crazy cool, right?
Okay. Lamination of the one Salama is it's 13 billion parameters is bigger than Mistral.
When you download them, you will actually see there many gigabytes. Okay. So the bigger the more gigabytes, smaller the better.
But even though Mistral is smaller, it is still better than a bigger competition.
And Lamarr came from Facebook account. So that's Facebook's own liberty.
There are so many of them you want to call the corner. I mean, again, you know, I won't tell you all of them, but there's so many.
So we call now is also something you can download another open source.

You know, I mean, this one is 90% of are pretty.
But then you can download record and run it on your own machine. Some are even small, so small that you can even run them on your phone.
Some are reasonable that you can train them on your on your laptop overnight can fine tune.
So there's a whole world of lamps that are not big anymore, but then they are like, acceptable.
You know, I live with that. I do it 90% accuracy. So then those can be paired with all of the rest of that, I'm telling you.
And we run on some kind of a blockchain and the magical magically will have
access to so much and you're starting to build like real world applications.
So please think in those terms, okay. And you can actually start companies, you know, as virtual Trojan,
literally start a plug and play turnkey system where somebody like a legal law firm, you know,
with all the lawsuits that they ever did, or a bunch of doctors that know so much about medical stuff,
they can pool all of the information into like a thing. And it's not your pretty store.
You don't put it up on Jeopardy, but you use it for your own internal use among your own peers.
It's a great idea. All right. So I'm going on ten grams, right? Ten grams.
I told you, Google just has all these ten grams that are so many words in order to have.
How many words? A few trillion. Yeah. So Google has 1 trillion different words that they pulled from so many web pages to the index then.
So then that is 1 trillion number of tokens talking to mean words.
Those 1 trillion words exist in, you know, so many, almost 900 million or something sentences, then those are so many unique grams by grams programs.
Right. So obviously, the more word combinations, the more these grams get.
In other words, you know how many sequences of five words, how many similar six words?
They're just some numbers. Very. But the idea is Google actually would give them all to you if you go there all or any grandma belong to you.
This basically was a meme at the time. Okay? Somebody in Japan made a joke about English that said all our base are belong to you.
And that became like a bad meme. So then that based on mocking that all our anger belong to you.
So they basically give you that anagram still.
And so you can even build your own search engine and then compare it to Google's anagrams and try and fix typos.
Okay. That is what all this is. So if you go there, you know.
Al and Graham belong to.
Ha ha. So I did like an engram search to complete the rest.
Okay, a little better joke there for you. Okay. In 2006, you know what a great day, right?
It is pretty neat to tell you exactly, you know. For example, you know, they tell you three grams.
Okay, a quarter. What are three words that occur together? Here's a small collection.
Wow. So ceramics, ceramics, ceramics serve us.
So the sweater slides, you know, the pictures came from anyway.
So you can then go in and actually look at all those engrams yourself.
I mean, there's so much in here, and there's no doubt they'll pair all this with multimodal.
It means they can pair it with images, you know, with radio audio in the future.
Okay.
So then all of this came from, you know, look what I told you so far that came from all these papers I don't want to read anymore the spell checkers.
And I can go and read this if you want. Before archive, we had this thing called sites.
Here was a paper server. Still exists.
Penn State. But no archive XIV is all but original sites.
Nobody has sites here anymore. But it's still there though. Great.
And then this. The paper. Yeah.
So let's do a Lowenstein, meaning let's look at Lowenstein at a high level and then more and more topics.
And I have so much to tell you. I didn't want to be perpetually behind. I want to try and catch up again.
So. Lowenstein First of all, is like a Russian name. Lowenstein The German version would be styled like Einstein.
Without H. Stein means a megabyte, and it could be a monk.
Einstein, Ina Stein, This one. Monk Einstein.

Okay. It's an ambition.
DEUTSCH okay. So at a distance here, the insert delete, substitute, insert, delete, quote.
How do you make how do you make one word? Any word becoming near the word?
All right. So this one tells you the example of kitten becoming sitting.
How do you turn Kitten into sitting? You go down the wrong path, but you don't keep on going because you can make it in the middle.
But that is not going to give you a sitting. So it's based on like locally optimal search.
I like local grid research is going in the proper direction so that it can get to the work that you want.
So if you want to go from K to S right, you would rather substitute K to as, but you get a word called sitting.
But it's not a word. Then sit and wait, then change it to sit in.
Now we have a home. What's that like?
I'm sitting on a park bench, so the E became an I the substitution and then you add a G this an extra like a pleasant local sitting.
So at a distance three because one substitution, second term, three executions, I can go from C into sitting.
I show you I can try this right now. I read all this.
Okay. They're not boring, but they're not that important. Really. And already told you so.
Idea is also you cannot do brute force, okay?
You cannot take it in and turn into every possible word in every possible way and see sitting was one of them or not.
It will generate so many useless words. Okay, so let's look at intractable problem.
In intractable problem if you want to go in the right direction.
Okay, so like this, like locally optimal service stations up all other words, you're going in the right direction.
Okay. Getting there. Okay. The others are dead ends.
Okay. So that is the part that are completely skipped. Really. So a little algorithm.
Okay, but then I can show you a visually. Okay.
I'm saying J by the way, a character positions in towards other words, it's like you're moving from you're basically moving this way.
There are two words with a position one position, two, three, four.
But each position I this index index, I is one word index, just another word and then compare I it.
So I just draw a picture an air for you. There it is.
Then given any two words like word on where to buy this.
Pretty funny. I think they should be chance to. I think you know like overall C language.
Okay. But except in C you don't have the word function. I don't know some words.
You're your intro strings. Okay. And towards basically then it'll return you like a distance 01230.
Because when the two words are the same, there's nothing to change everywhere.
And there's at a distance of zero or one or two or three. So I'm going to show you some, a couple of pretty cool resources.
Second one is better than the first one. So this one, how do you go from fried to fresh, for example?
Okay, you would change E you would change it as you would change, for example, in orange today.
Okay. So will not read through all of them. But it's very cool though, you know.
So speaking of all languages, you know, well, pretty hard.
But there it is so fried to go hard as it become fresh.
If you read through all of this, I want to tell you it is one based index, meaning this is called position one,
position two, three, four, five, position one, two, three, four, five, six, seven.
It can keep going. So the dark part is going to be zero.
In other words, usually array starting number zero. Here this array say index.
Remember one Fortran is a critical language, but array index starts from one.
So Loewenstein probably knew Fortran. Okay, so it didn't matter. But then I tried it usually.
So like this, right? The index actually starts from one, so zero is used for nothing.
Okay. So this is the part that I would love for you to read yourself.
Okay, so it's going like this. In other words, if does not have to be changed.
Right. Likewise are also does not have to be changed. That is why f remains the same throughout.
Can she fill this matrix at the very end? It pops out at a distance and on the right hand side, the bottom right value would be at a distance.

Okay, So again, and I'm purposely skipping because it does some fun when you learn it yourself,
I'll show you another paper which is even better than this. This one. One more example to medium article.
Oh. Oh, okay.
Okay, she said, according to Al Gore at the meeting. You know, you go from right to left, sort of left, right.
It means, you know, some words like one would like words like this. So this support And what about this support?
What about the support she picks in smaller and smaller parts of that weren't that towards So that is your reaction And then yeah,
this is the part that I enjoy once again. So please read not for the exam.
Okay, but just for your own sake. The student also had a function is a function transforms input output.
No explained what a function is so that you already know all this.
Okay. You also know piecewise function. Piecewise function means if statement functions.
If this do this as opposed to y equal to x squared that is not a piecewise function is a continuous function continuous.
And so what function this derivative sort of way. Okay. So then ultimately it comes down to this.
Okay, it's very neat. Okay. Right. This is like monitor is going to people.
So I want to read it now for you though. Okay. Here's a one letter change.
How do you go from cat to cap? Just change T to be at a distance is one.
Okay. And then a more bigger example. So Kitten, the sitting, the one that I showed in the slide is right here.
So please spend maybe a half an hour if you want going through this, and then it'll tell you exactly how it's happening again.
You know, you change some letters, but you don't change other letters. Yeah.
Great. And it tracks how many changes and history, as you know. Great.
So. Lowenstein And there's even more media articles.
Many people want to explain this to other people, but you can have a tragedy to explain it if you want.
Okay. Okay, Let's run this. Okay, kitten.
Actually. Yeah. Backspace.
Let's change that to salt. Okay? Why not? Okay.
Salt. Okay. See that quote.
Okay, so two changes. Instant. So then I'm just asking you if you want to actually explain still.
So this is just a very best way to understand. Okay. So that it tells you why.
Like, why did this one come from a no? What is one come from and do?
So if you do enough of them, then you will know what is going on.
You don't need to know the answer. Or you can read this article.
What about the whole Kevin sitting business? What about changing sitting the kitten?
Sitting. Let's do it backwards. Oh.
Obviously symmetry in order. Doesn't matter which way you go, right? That's actually pretty cool.
Hopefully it will be a different type of impact algorithm. Okay. So, okay, what do you call that?
Property when order doesn't matter. This algorithm has a very interesting property.
An across product across B and B cross are not identical.
In fact, one less 180 degrees competitor OC one is in minus stellar.
Where do you call that? Yeah, commutation Exactly.
In this case, this algorithm shows displace committed to property order doesn't matter in cross product order matters.
Okay In that product order doesn't matter. Hey cool.
When you cross this in metric multiplication order matters.
Sometimes the opposite might not even be mathematically possible. Okay.
OC can only do a Crosby, but not because it quote because you know columns and those have to match.
Great. So this is a very neat outcome I should just move on to more to tell you.
Yes we ran that Oh yeah it's fine that you can also run this.
I made a virgin after all. I always make some versions.
Okay. I look so small. So you can always do a little business.
So here I am showing you the diagonal matrix, like in a straight line. You can respond like this however you want, but there it is.
Equal weight makes it tend to stay. Huh? You can change it into salt if you want.
Okay. And so they're centered at a distance of talking. And the algorithm is right here.

Again, you can go look at it if you want. JavaScript.
Again. There it is. Right. Function loewenstein string on seem to.
There's no return to simply pretty misprint. Okay.
All right. Cool. Yes. He said stay so stay good doubt stay when you guys email me once in a while somebody says Hi professor stay.
You just makes me laugh. Oh by the way when I write back to them.
Actually fixed that typo, but I don't tell them I fixed that type of magical effects it.
I'm not still okay. I'm not leave either. Should I stay or should I go?
Okay. Oh, wait. There might be more.
That might be more. Control. Zero. Yes.
This is cool. There are so many movies that have been made in Hollywood and tried to look at Bollywood.
You know, try an order ordering. So what if somebody wants to search for a whole bunch of movies and they start typing something
either correctly or incorrectly and then say incorrectly for it for business purposes?
What if you do learn stand distance editing on the misspelled word letter in this case and find all the actual movies that how to properly spelt word?
That's a cool little thing. Okay. You're going to discover most. Okay. So then this is a paper that talks about how that is done.
This is a very neat application allowing some unusual application because usually you would do it in like a word processing or text, whatever.
Right. Or maybe the search engine. But this one is spell correction using Lowenstein for movie name discovery.
Like I want a Google movie names, but then I might not know the way to spell something.
And then how am I going to like this whole flowchart right there?
You can go it afterwards. And the idea is just it's not important.
But the whole thing is where the movie names you and comes from the actual real movie names.
You know, it comes from IMDB, the big, you know, Internet movie database.
But then show you just the bottom line here. Okay. So suppose somebody typed for the world through the word through.
They instead are typed through. Go here.
There it is. Okay, then throw matches through Lowenstein.
In all of this, you know, I mean, we had the world saw that was going to come on our turbo and ask DreamWorks,
for example, Al Capone, who he writes for Turbo and then Troy, Helen of Troy.
They all match. In other words, throw can be easily converted to all of them.
Then they're all potential matches that any movie that has to work in. This has a word.
[INAUDIBLE] pull out the movie name for a joke. Then here they explained Lowenstein one more time.
It's pretty neat. Okay, so you can have some custom search engines and either do this for you.
Last but not least, do not turn to a next topic.
This notion of at a distance like we have here, but waiting them some at a distance as if they happen there,
I'd say, and make an added Tamika one change and I get a new word because of that.
Depending on what that letter change was specifically eg or change to be,
you know, and zig or change to why some of those changes should be rewarded more,
meaning they're scaled higher because they are more important changes, meaning many people make those type of all the time.
Some other changes, even though they are legal, they led to a new word. They're not that important.
So you're still them less. Okay, that is called a weighted at a distance.
So not already sort of the same. So what I go for what it would be in three different edits and so far I said at a distance of three.
But no, I might no longer say it is three, I might say one times 0.9 plus one times 0.2 plus one times 0.01.
That is maintain a distance because even though I made three changes, one this 90% important, the other one is 20% important.
The other one is, you know, 0.1% important.
So that way I can have different possible changes from any letter to any of the letter, have a weight associated with them.
See, literally take A to Z, A to Z, you see higher numbers.

For example, this one E and F are quite commonly transposed.
So if you if you change A to Z and at least a new word that is so highly weighted,
okay, So the bigger the number, the more important that change, so to speak.
So it's a pretty neat table to form. A table only comes from someone like Google who has seen so many Misspellings.
So can they know exactly what where to correct. They compiled this kind of a statistic.
It's a pretty fascinating statistic. This reflects like what typos people actually make.
Okay. That's all. Hardly anybody swaps, you know, like a misstep, like H versus V.
Okay, then that's literally zero sum way to zero. Yeah.
Okay. I know the exact opposite of zero to simply underline the higher ones.
That's pretty much it. Oh, yeah. By the way, in machine learning, as you know, you can also call this a confusion matrix again.
So the middle is no change at all. A's not true. A's simply change to a B since the B.
Yeah. So confusion happens here in machine learning.
It is all about things like false positive false negatives. True positives to negatives usually make like a square, right?
And to say you actually want true positives to be classified.
If positives are classified as positives, you call true positive for negative is classified as negative tornado.
That's not a problem. You want the machine to do that. But the problem happens when you do the opposite.
When you classify a negative as a positive, that is called a false positive.
Take a COVID test. You actually don't have COVID, but the machine says, Oh my God, the line, Oh no, that's called a false positive.
If somebody becomes pregnant, false positives is a scary thing.
You know, you didn't mean to get pregnant. Okay. But then the pregnancy test person says, Oh, my God, you're pregnant.
What now? It's pretty bad. So false positives.
Can you train for a super false negative is actually even worse.
In false negative, you actually have COVID. But the test, the PCR test is pretty bad, right?
PCR test. And then it says, you know, you're going to infect everybody.
It's called a false negative. So false positives and false negatives, they become part of confusion.
So in a way, it's like these are all false positive, false negatives. The missteps.
Okay, they're spelling errors. Cool. You know what I'm talking about, okay?
In a binary classification, they can make a very simple table. You can have an multiclass table know.
Then you're going to have the number of instances of A classified as B or C, SD.
You can make a whole matrix of them for a multiclass classification,
but in the middle always be zero because you know, that's the same I'm confusion matrix.
You know, you've seen all this, okay, this that's what I'm talking about. So true positive is good to negative is good, but these are bad and so on.
Great. Yeah. Okay. You can do stuff like that.
Listen, I mean, she in here, you know, if it's classified as a that's good.
So hopefully these numbers are pretty high. That means they have high accuracy.
But then, if it is classified, I see that so many instances and so on.
Okay. By the way, these don't have to be symmetric. Okay? If A's misclassified ac5 times, C might not need to be misclassified as a five times.
Also as an asymmetric matrix there. Great.
So the connection, that mission learning. All right. So then we have that.
With that, we're done. Wow.
So, you know, the take home message in all of this is things like and Grams, the learning to understand some very basic ideas.
The rest are all just simply detail you guys. We don't need to worry too much about it.
Okay. Let's then switch gears and talk about question answering, which is even more relevant than autocorrect.
I mean, autocorrect will always be with us, nor as long as we type.
Someday in the future will not type.
Everything is so audio based on a pretty you know, By the way, Jeopardy four can already process audio in ways like Alexa Siri, you know.

But I think typing will always be with us because you can actually type something.
But if you're typing, you're going to make typos and you have autocorrect.
Okay, Q&A, You know, this used to have more slides, but I remove some of the extraneous ones, you know, because there's too much.
Okay. So again, here are some pretty core ideas. Okay.
The biggest core idea I can come out and tell you right now, you ask questions in Google at first.
And when Google first came out, there were no questions.
Somebody would just type towards Los Angeles and then it'll pull out all the words with the word Los Angeles in it.
Literally a tough idea for game and page rank it, that's all.
But these days, people will say things like, what is the average temperature in Los Angeles?
Over here are what is the crime rate in Los Angeles? How can a flight to Los Angeles, what kind of car rentals are not available in Los Angeles?
So it's an actual question people ask.
Then the answer has to be like, your answer the question so hard, ask the question, turn into an answer that cannot be a bigger use for an L.A.
I mean, when you pretty came along, that's exactly what happened.
What a way to approach actually a question, You know, like write me an essay about God.
You know, that does not sound like the question you're asking. Can you please write me?
You know, and I say word God. So the whole Q&A deal took off when L.A. came along.
So that is how it will be for a long, long time.
But immediately what happened was, if you only used to allow myself to answer questions, it is like, you know,
like one of the very first demos that borrowed had completely failed because the Google people went to friends, okay, and had the big Bard opening.
So Bard, by the way, was second tragedy to us first.
So Google got suppressed and said, Oh my God, in an open air, [INAUDIBLE] are beating us, you know?
And it was something they came up with. Bird But Bird had in their own opening press release in France, a big mistake.
It talked about exoplanets. The question was one way exoplanets discovered so barks or something like in, you know, recently to turn this green.
But the reality is if you go to NASA pages, it goes back to the nineties.
Okay. And somebody didn't catch that so I saw embarrassing.
I should share that deal then. That's how important this has actually become.
Bard Press release. Press release error.
When that happened overnight, Google's evaluation went down by $100 billion.
Shocking. Seriously shocking. People say what?
Okay, now let's solve by opening our stock. Can you imagine overnight the valuation being wiped out?
That much was so bad. Hundred.
2 billion was off by 2 billion. Excuse me.
Crazy head. Yeah. I mean, this looks so bad.
Okay, this one error. Well.
Okay. Yep. Yeah.
I mean, look at, you know, Right. Punish.
Okay, let's see what she says. I don't know. Hmm.
If I find the one little animated animated just sort of in such a hurry to go to friends.
And I noticed a big board right there made a differentiation. Okay, with words in it.
Somebody who's a fact checked it. Yeah, Google. After all, you can Google your own damn question.
And they're going to cut it. But then they were too arrogant or too, I guess, you know, caught flat footed and.
Yeah, did Google board make an error, you know, like all of this sort of stuff.
I want to go to image search. Okay. Yeah, this one.
Okay. See the. Yeah. This one.
Oh, my God. Okay, so James Webb telescope was really selected no less than five years ago.
Whereas NASA's previous satellites found exoplanets a long, long, long time ago.
But then here's what the chat board said. Okay, This is correct. There's also correct sort of one wrong.
And that is truly bad because a little kid will have no idea how to get.

No. You're lying to a kid. So, you know.
James Webb telescope took the first pictures of our planet. It did not occur in the future.
What'll happen is if we let this be, people get tired of pointing out errors.
Okay? Astronomers would say that is not true, but we have no time to go and correct graphics errors.
I would basically say f this, I give up. So if I say that and stop correcting them, then you will suffer because you don't know what I know.
I see there was misinformation spread. So thankfully we have a solution.
Thankfully, in this case, what this could have happened, except when they did it, there was no what I'm going to tell you.
There's something called retrieval augmentation. That is the magic fix, retrieval augmentation.
But civil with the IEEE Business. IEEE will not receive a civil augmentation.
Mr. Magic idea said ritual augmentation. You don't let the church obedience or anything, you tell it.
Your job is to understand my English query and then go to an external database.
In this case, all of NASA's database, factual database, where they have everything about actual planets going back to timelines,
actual satellite names, you know, who was a project manager, whatever the [INAUDIBLE] you want to know.
Go get the answer from there. If they had done that, this would not have happened.
So we're going to look at work records. Then we can get back to kind of large language models.
They are everywhere. They get some things amazingly right and other things very interestingly wrong.
My name is Marina Milewski. I am a senior research scientist here at IBM Research,
and I want to tell you about a framework to help large language models be more accurate and more up to date retrieval, augmented generation or rag.
Let's just talk about the generation quiet for a minute. So forget the retrieval, augmented generation.
This refers to large language models or elements that generate text in response to a user query referred to as a problem.
These models can have some undesirable behavior to tell you and Adam to illustrate this.
So my kids, they recently asked me this question in our solar system.
What planet has the most moons? And my response was, Oh, that's really great that you're asking this question.
I loved space when I was your age. Of course, that was like 30 years ago.
But I know I read an article. I read the article, said that it was Jupiter and E.T.
So that's now actually there's a couple islands here just plexiglass screen right there.
By the way, if you didn't know. No source. And the error made up the answer.
That's where you go for personal information about how long the Don't ask tell one because either doesn't know or ordered it with largely those.
A large language model. Okay. So so far, that's what the user wanted, right?
If you then ask, tell them it might tell you something incorrect.
But then we need to take this and then ask an external vector database or a knowledge graph.
There are two different ways to augment to put in like factual knowledge.
One is to make a knowledge graph from last class, an actual knowledge graph,
and have that knowledge graph be searched for named entities and our planet names satellite names in order.
Okay. Or you can make a vector database, which I haven't told you,
but it's basically numerical values to turn all the facts into a python list of numbers and say there are 50,000 numbers.
The python list has 50,000 floating points each. Floating point number is an axis and mathematical space.
First number you plot on one dimension X dimension. Second number you plant in the Y axis, third number on the Z-AXIS.
Then for Texas. For Texas. So you have 50,000 element long list has become 1.1 dart in a 50,000 dimensional space.
That one sentence of actual something about our exoplanet. Okay.
The second sentence of the exoplanet also n in the same political and dimensional space, it became another point.
So every row, every sentence, every training, every input data basically becomes a da da da da da in some multidimensional space.
Okay, that is the input. In other words, that's where the high quality knowledge lies.
Then when you ask about exoplanets, your query also becomes a vector that is also 50,000 tokens long,
and that vector will also become a point in the same practical and multidimensional space.
Now you do a proximity, search your new query, look around within a radius of ten,

whatever number you set, what other data that exists, and those are facts, okay?
And then you pull them out and you turn them into the extra part of your prompt.
Your prompt for the user was, you know, what planet has most moons.
But now that prompt is augmented. Delisle doesn't know that. Okay.
But now the prompt is a much bigger prompt and the argumentation is the actual answer that you want.
And you tell them the system to tell them. Now spit it back to the user.
So now the user gets a factual answer. It's a great idea, right?
Such an amazing thing to do. Would confidently say, okay, I have been trained and from what I know in my so that's bad.
Okay, the answer is Jupiter. The answer is wrong, but.
You know, we don't know a lot of language models very confident and want an answer.
Now, what happens when you add this retriever retrieval augmented part here?
What does that external database like?
Some collection of documents.
So this could be documents like PDF files know that have been that have been tokenized and turned into vector embeddings.
Okay, not raw PDF files. There's not anything you start with writing anything, but you turn them all into numbers.
There's going to be music you can search for music like Shazam, Whisper Something, a song.
It'll pull you all the songs that are pretty similar to what you're you send image search.
I can take a photograph and say, Show me all the things that look like that. These are better than sequel because you cannot do sequel queries.
Okay. How do you describe a photograph? Suppose I'm carrying this and you've never seen this coke bottle before.
Well, what is a citizen to send? Take a little picture from any phone camera angle and say, What is this?
Then my pixels will become again a point in the multidimensional space.
But hopefully the system already has many bottles somebody already added.
Then it'll come back and tell you, Hey, that's a coke bottle. It's a cool idea, right?
It can search for anything. Okay. So likewise.
So this can be a vector embedding, which is a multi-dimensional point or can be a knowledge graph of Jupiter,
13 moons discovered by Herschel, William Michel and so on.
So you can do either or you can do both, actually. Policies, whatever.
The point, though, now is that the line first goes and talks up.
That is where the good answer comes from. Hey, can you retrieve for me?
Information that is relevant to what the user's query was. And now with this retriever augmented answer, it's not Jupiter anymore.
We know that it is. It's super cool. What does this look like?
Well, every company in the World Bank of America, 76 gas station, subway, 7-Eleven,
U.S. wants this because Chad Bartz, the current generation Chad boards completely suck the keyword based.
Okay? So if you ask something that's not in the cure, it basically says, Wait, I'll get my supervisor.
It sucks. But with this, anybody any company can take all the knowledge that they want the existing.
And I have at the middle of the night and make high quality databases like that, you know, and then how the chat bot ecosystem.
Then I can at three in the morning ask Bank of America how many different kinds of checking accounts do you have it?
Can you help me open one? It'll actually do the right thing. As a boss, Lang Lang has gone to one result.
For me it is very cool personally because I worked on a very large knowledge graph project called CIC Psych.
It's called Expert Systems. Okay, so now experts systems are back in full force and they'll came and killed.
Experts estimate multiple said expert systems are very slow. They're not scalable, you know, painful.
So I'll throw data at the problem and we'll figure it out. But data basically sucks because at least, you know, your tail tucked between your legs,
you'll come back to expert systems because it is high quality.
But now you get the best of both because you can chat in natural language with the expert system.
So they think, okay, it's a cool idea on the LAMP framework or the model app with.
So one last thing I want to tell you. So how does this actual retrieval happen and how do you add it to the prompt?
You would need a third component. It's called length chain.

So you have Al-Alam, you have the external database, and you have this thing called long chain length.
Chain is a prom programing language. Okay, So long chain can then take this prompt already and then go in here and search and come
back with the result added to a make a bigger problem to prompt augmentation system.
I actually forgot to say something and I made a little note here. I'll do attendance in a few minutes.
But we are just going, letting. The driver model says, okay, okay, this great.
This is better than what I can explain this overnight. Here it is. My response.
But now in the grand framework, the general model actually has an instruction that says, No, no, no.
First, go and retrieve relevant content.
Combine that with the user's question in the prompt, you can actually say, Do not answer the question yourself.
Go consult an extra source. If the extra source has no answer either come back and tell me.
I don't know. You type all that into a little kid, okay.
And then that can actually happen because not everything that you search for is in the external database, right?
Then it'll come and tell you. I don't know. It's better that I tell you.
I don't know. Then let you and only then generate the answer.
So the prompt now has three parts.
The instruction to pay attention to the Retrieved content together with the user's question.
Now give a response. And in fact, now you can give evidence for why your response was what it was.
The evidence can come from here so that hopefully you can see. So now the next need because it is solid.
Very cool stuff. Okay. So I just wanted to show you this notion of actual augmentation.
This explained in so many different ways. Again, you spend a few minutes, you will get it.
Okay. So the idea see this in the past, meaning until the act came along, this was the only chain.
So whatever the heck tell you it's split it back at you, James, for telescoping in exoplanets for the first time.
But now the national database can tell you that is not true. Then you override what Talleyrand says and you get a better answer.
Great. So the thing that I forgot to tell you is this Go to this incredible site called Deep Learning that is founded by Andrew Indonesia.
There's about 30 different videos in here. They're all courses, right? Do every single one of them, they're all free.
Are free because you can freely audit them.
If you want to pay money for bucks, they'll actually grade your homework and give you the results back to you.
This is literally gold versus actor database, you know, and then prompt engineering, that is basically the long chain from programing.
And then the more prompts to write. Okay this guy wrote Long chain.
His name is Harry Chase's inventor of Long chain.
I'm talking about this new short course online chain for l m large language model application development.
This is built in collaboration tools like this, creator of long chain arms,
have made it possible to use, prompting to develop powerful applications much faster than ever.
Small, but an application, say building a question answering system to ask questions about the text documents
that might require prompting and telling them multiple times with multiple inputs,
parsing the applets to then feed it to downstream problems and so on.
And so this is all the code needed to build these applications that chain initially,
that's what launching was born for, not for external augmentation at all.
It is simply to make the prompting is prompting can be a multistep. You can then code all of that.
So why you know from similar things over and over basically automate prompting again but people suddenly discovered, wow, I can do a much better thing with this.
I can actually make the problem go out as a now land chain has something called functions launching functions.
This a brand new thing two weeks ago. And so then you can start writing more things.
You can write agents. So an agent is basically imagine freaking LDL alarm limits 1 to 1 alarm.
Imagine making ten different copies of the alarm.
They're all running in parallel features and each one is answering with a different bunch of problems.
You can they're all doing different things. Here are people you tell one of them your own marketing campaign is other one.
You're in advertising, you report on revenue. You know, you don't invent new products.

You did according to the documentation and all of you work together.
So that is a crazy evolution. You know, this basically, like magically many people collaborating, doing work.
Okay, That's all now possible. So that is when this is taking off in crazy many different ways.
So one of them would be the whole to love lamentation. So I'll actually show you that it's here as well.
So please do every single one of them. See select right there. Okay. That is external data.
That could be a spreadsheet, PDF file, text file. You can ask a question looking through that or 80 different loaders.
So that means you can read all your PDFs. They're like data connectors.
Okay. Is suddenly become about databases because databases can hold a high quality knowledge.
Okay, fine. So fine tuning is the opposite. Fine tuning means you don't have external sources.
You take your own alarm and basically train the last layer, so to speak.
Okay, fine tuning. It means going to grad school.
So undergrad computer science is basically overall like, you know, you're, you know, a little bit of everything, right?
But now suddenly go to grad school, focus on cybersecurity, focus on software development, focused on game design.
That's what fine tuning is. So it's a little bit different. All right. So this just goes on and on and on.
This is great. Again, you can do search beyond Google keywords, okay?
I mean, and then you can build applications. So now the whole alarm has become a whole platform.
You know, that's actually what it is. This about image generation. In fact, that's a definition that I told you, you know, stable diffusion.
This is like so critical. Cool. So please look at that function.
This look just a few days old. I'll just play. I'm thrilled to be back with Harrison Chase, founder of Nightshade, and instructs her on a new course.
Functions, tools and agents. This is mind blowing.
It's great to be back in early causes. We shared how to use launching, including how to use it to chat with your data.
In the short time since we treated those causes,
there have been significant advancements in those and the libraries to support their use as a developer too.
This course was created to update you on functioning coding, such as Open the Eyes, work on letting alarms,
how other functions the their cell alarm blew the doors open and let them open and they have API or just an open API.
With that you can run or you can type something and send calls to MATLAB.
Suddenly you can say, Here's an email, share some of the emails, multiply them together.
Those are MATLAB commands or whatever dot m file. Right?
But now you can do it by chatting because what the long chain thing can do is pass what you typed and turn that into actual MATLAB calls.
And how does it get to MATLAB because of the plugin API.
So plug in for open AI for the for makes it too pretty for connectable to any other software application in the world.
PowerPoint Excel. You can make spreadsheets just by typing, make a spreadsheet, fill the first row with normal distribution.
It'll actually do it for you. Okay. Wow. In 3D graphics, you can go to Blender and say Fill a room filled with spheres of radio between ten and 100.
They'll make 3D spheres for you. So no more clicking buttons, because every button you click and blender a unity.
There's a python command for it. Then now you can run those commands by typing brain text.
Okay, That is what you're talking about. But now you can do that too long. Chained to the U.
And more things to automate translate to be very useful for handling structured data.
Something that has struggled with you because you can run sequel commands servers.
So you can run sequel command. So you can actually you can type something natural language like in know in My company.
Find me the top 20 employees that get paid the most salary.
So it'll do a salary sorting. It'll basically go on the salary column and then pull out the top 20 and give the names to you.
Those are classic sequel joins. Okay. But now you can type and actually do it for you to work in natural language for the rest of the computing world.
Works with formatted things like function calls or APIs.
Want specific data in specific format. You can even do rest calls, by the way.
So you have Rest API that returns stock prices.
Now you can actually say, you know, so what is output talk work today and give me the stock distribution for the last

week.
Plot output stop distribution because it makes API calls and incense sends date last week's date.
What gets all the numbers and plots it for you? Basic training Williams can now understand and output data like Jason.
And in this course you will get a chance to work with this directly.
These updates make alarms more predictable and reliable, as well as being better at understanding what to use tools.
This also means external programing in Photoshop or blender reasons about how to use tools or Oracle.
The step problems in this course will cover several things.
First, we'll start by explaining some recent advancements in an API called Lane Train Expression Language.
Yeah, that's brand new. Wow. So here, chest put in.
It looks like a regular expression, but is basically prompt expression. Language.
Chicken. Some structured like regular expression, but for prompts.
And those can be then parsed and substituted with actual data so you can automatically generate like more problems.
Okay. It's all about automating problem generation. I mean, this is getting so seriously powerful.
Elam has become like an operating system. Okay. So imagine.
Elam operating system. What will you build on top of it? It's basically a question.
There's so much in here. Also, I wanted to hear, you know, when people come to my office, I play 2000 companies and nobody called me back.
Screw all that. Screw the fang companies. Okay? Why are you going to make them do this?
The jobs will come running to you, watching to tell you so.
One more. Yep.
Okay. You know all of these, right?
Actually, you know, this is largely l rag jobs.
I mean, it's that simple or it's real lang change jobs. Okay, See that?
So that right there, there's so many okay. So many different unintended elements around I can show you L and lang change jobs.
This is truly where everything is. Jack Lang It was just such sadness, so many jobs actually available.
You're looking in the wrong place and the deep learning that a can quickly.
I forgot to tell you. Okay. So once you sign up, it is not just watching him talk.
They have like Jupyter notebooks and you can actually click on the code and four can run it.
You can run it and teach yourself that, okay, how much better can I get?
You don't have to install Jupyter notebook call nothing. It runs on, you know, like a long chain that come from a, B and deep learning that I.
So it's crazy, right? Amazing. So now we can do a Q&A quite fast again.
All right. So again, this notion of question answering. You no longer do keyword search.
You actually ask somebody asked a question like, you know, what is your most popular course?
Somebody would tell you real estate. You know, your most popular measure, by the way, is real estate, if it is, none of that.
Yes. So we got previously document retrieval. That is keyword based.
But now you need to actually answer questions. This is how you answer questions.
You do like information retrieval from a lecture database, probably knowledge graph.
And even based on what the user asked, they tend to ask the same kind of question.
So then that should be factored into answering them. Also, like where does the pizza place near me?
When you say near me, your wife needs to, you know, get your location and we'll get to pizza place near you.
Right. So stuff like that, Right. So many other things that are now used to do Q&A.
It is not just simply a keyword search. That's that's what the slide says.
Okay. This cannabis in a way. So people ask all these things, Oh, please turn that can have that.
No, Come on. Okay. Okay.
We can be very snarky and sarcastic and say if you need to ask it along there, you probably should have kids.
Please don't propagate. Okay. I was just looking for square die family.
That that'd be a pretty crazy example, by the way. The sad thing is I'm going to tell you.
Okay. I'll tell you if you say how to commit suicide.
Bar. One answer Chargeability. Also an answer grok will answer.
And that's pretty bad because when somebody asked that it should not answer them, you should say, I'll get your help.

You know, don't do that. It's pretty horrible. You know, Please don't do that. Not answer them to grok this will do it.
And Elan, like an idiot, would go on the press and say, Oh, it's not my fault.
Why did you ask that? That's a stupid thing to say. Okay, so basically throw the responsibility back to user.
What the [INAUDIBLE]? Right? Come on. So no, no, no human would answer that, okay?
Unless you're cruel. So that's actually what's going to happen. All right, So, you know, what is the meaning of, like, 42 plastics?
The so, like, pretty cool, right? Yeah. You know, how does Google notice these actual questions?
People ask them? Google just got to mind the logs. Wow. Amazing.
Then you know, I can skip some of these. Okay. Again, you know, ask who was the prime minister of Australia?
Now, hopefully can answer that. Okay.
It's like small stuff, you know, again, in a host, I don't really want to Typekit, but if you want, you can try it.
Who can ask that to a video game? Who was the Prime Minister of Australia during the Great Depression?
So that extra part matters. Without that, I'll give you the current Prime Minister.
But the Great Depression. So go back to 1937 or even the US President.
Okay. As called. Yeah. Then the point is with things like snippets, when you say how old is Mariah Carey?
Okay. Obviously asking for an age. Right. Which it found. But because it is Mariah Carey, you also probably want to know more about her.
Right. That is a snippet. So it even gives you some of the albums and spells and like all that.
And you can find where you can find her albums. And there's the answer. And sure enough is 53.
In other words, to making your search your question. You know, that's your question in a better way.
That's all. Which is how it should be. And nobody can go even better.
Okay. By the way, this all plea bargain, in a sense, but it's not the charge.
If it is not the first one to do the whole Q&A business. Okay. Google had this all along.
But now the question can be more an informal, like an actual language.
That is all when you can have what's called context.
You can continue the conversation here, even though you ask the next question, What do I should born?
There's no connection between your first query and the second query. The search engine is serving like many million queries per second.
It is not holding on to what you asked again, whereas in the Al-Alam call history context,
it basically remembers all the things you asked and maybe the next question is going to answer.
That will be based on all the previous things you asked. You know, that's that's pretty cool actually, after context window.
So the context window can be about a few thousand, 10,000, 20,000, 30,000 words.
Beyond that, it doesn't remember what you ask, so to speak. Again, that is sort of 83, 83, all that was born.
But now the context window is up to a few hundred kilo, 100, 100 keywords.
Context is getting bigger, but there are even more bigger ones, you know.
In Stanford, there's one model that context can have 1 million words, literally 1 million words, but maybe you can remember 1 million words.
Basically, you had a conversation with that in the past. 1 million words is how many words we used during the course of one week.
It means on a Sunday night it can even remember what you said the previous Monday morning is so powerful.
1 million words. Wow. So then that is all part of this Q&A business.
Okay. One m context window.
Context window Stanford. Yeah.
Yeah. Look at the 101 million tokens and beyond.
Oh. There's a new transformer called R.A., so you can look up the R.A. Transformer and then you would know.
It's crazy, right? 1 million contacts windows have.
Back care again, you can say, how tall is Mount Everest?
That is the factual answer. 29,000 feet. But now it gives you a lot more.
Give you the best campsite in our other mountains nearby. And this is in the US, Mount McKinley, Alaska.
Give you Mount Everest and more facts about it. So, you know, the answer is getting more and more interesting, right?
And then in the meanwhile, the actual answer is still right in the past.
Maybe only give you that number, but now it can give you more and more.
The question is how you know, one one answer before Alison's knowledge graphs towards knowledge graphs.

So knowledge graphs of how that can go on, find the word Mount Everest, and then it says Mount Everest.
You know, you can go from Nepal, our tallest mountain in the world, you know, bordered by China, all kinds of things.
Right? And then they'll come and tell you all that. So knowledge graph is the answer.
So now it lands. Knowledge graph is still answer. But previously we thought Al-Alam, Google's Internet search engine, had the job of taking the knowledge graph entities, which are simply wires connections.
Right. And turning that into English. But now the Al-Alam can do that for you.
So that's really better than what Google has been doing so far, says Bartok.
So in 20. I'll tell you what, don't get up and go, you guys,
because I'm going to call your name and at least wait till I call your name and then go minute or no, that tune me out.
Well, if the name shows up to where I give you a change a little to give you a chance.
Okay, so then who is it? That's funny.
Okay. V.S. Naraya V.S., I would say I'm going to guess it's V.S. Narayanan.
So if you're here, please, I think you're wonderful. P, vijay and P Return.
They might be in the library. Okay, then please do a little chat or send me.
Send me a picture. Okay. Do a few more.
Russian are Russian or Russian are.
Yeah. Yeah. Cool. Yeah. You're more happy than I am that you're here.
Okay. Geography, Congestion. Yeah.
Cool. Okay. Thank you. Grandmother.
There's no going to. Norma. Norma.
Yeah. Yeah. Monroe. Mount Wilson, I think.
Mount Wilson. I'm sorry. It's empty. Wilson time.
I apologize. Okay? There's no Mount Wilson. There's an observatory behind.
Oh, yeah. Sorry. It's not. It's not mount, right? For sure.
It's not Mount Wilson. Hey, Lone Star.
Mount Matt. Sorry. A little joke at your expense.
Sorry about that. Matt Wilson. Not oralism chin chin.
Chin. Chin chin. Great. Hey, let's actually quit while over our head.
Wait. Yeah. So, yes.
So what Google did West is okay. When you have a question, somebody asks a question like, where's Los Angeles?
Okay, hey, look, east or no, I should do it again.
All right. So they do it in Michigan. Yeah, yeah, yeah, yeah, yeah.
Hey, you know this zero sum game? Have you noticed that this whole course is a zero sum game?
It means you're somebody's score goes down by five, You know, all your score goes up a fine.
How is that? Because it's all relative grading. Okay. Oh, wow.
Okay. So call him. Call his name. Keep doing it till his name comes up.
You guys are mean. Okay. Oh, my God. Okay, So, uh, yeah, there's simply search for your question.
Literally. Go on. Search for your question, but not all questions directly.
Have a page where the question is directly answered. I mean, because magic is actually that.
Okay, semantic search, I might say. Give me articles about biology.
Somebody wrote a paper about cancer, which is clearly biology, but they don't have to say cancer.
This part of biology, this whole cancer paper, now they can bring it to you.
Okay, so are beyond keywords and that is magical.
So then that kind of a cancer paper would not be given to me if I ask questions like, Tell me about biology.
You understand how that works. If you if you search for the question verbatim with no change in a document,
you're not going to get all the answers will only get a part of the answers. But that's what they used to do.
And then when they find the actual question, you know, like what is life?
Suppose what is life then? Maybe Schrodinger was a quantum mechanics guy.
He's also philosophical. So he wrote a paper about life. He said, What is life?
Well, life is a collection of molecules that self reinforce each other.
So you find that document that literally says, what is life re a section of a book?
And then the next sentence that answers the question, Google will give that to you.
That is how they answer questions. It's a pretty weird approach that kind of works,

but it'll fail if the question you type is not actually in any document because then they cannot find an answer.
And even if they returned the very next sentence, maybe the sentence was simply an introduction.
You know, life is a very mysterious thing. And then they go on answering afterwards what life is, right?
They would only give me the first sentence. Then, you know, if I ask what is life, then Google are tell me life is the most mysterious thing.
I didn't want to look. I actually want to know where there's so many problems with this approach against what I'm saying.
So all works poorly most of the time. Yeah. Okay.
Unless it is like Jeopardy! Okay. But Jeopardy also is wrong because in jeopardy, no answer followed by question.
You know, then even Jeopardy questions won't be sent to your account is weight problem.
But this tells you, Larry, and on this search that people just like you and me, they tried to most obvious ways, okay.
But people still kept using them and they became like more and more rich. Right. And then they basically fixed in on their approach.
Algorithms turned out to be perfect the first time, you know, going to start with something that doesn't work, really.
But over time, you hire more people that are very smart and say, Hey, fix this.
Okay, so here's what they do now. The US knowledge graph and also the question itself.
You can tokenize it and remove the extra words like what and all that. Right? And go and look in engrams.
Okay. And the question is no one engram So you find Engram that is part of your question.
It can know how many more say Jupiter, how Jupiter moons. Wow. Cool.
Now go and find our Jupiter moons in any document and hopefully that'll be the right answer.
Okay, I'm pretty to summarize what you find. You learn that this way, the whole structuring of knowledge comes in.
So if you find the word Jupiter, then in the wording it it says Jupiter is a former planet.
Planet is like a astronomical body.
That's part of the universe, you know, then they can use the whole chain up the hierarchy to answer a question because of a knowledge graph.
But that is a pretty big knowledge graph. You know, go. Which this from Princeton.
Or if all else fails, you can start today now.
That is why the new bard, all of that comes in to try to really understand these are all basically tricks, is what we're saying.
Okay. All tricks. Forget tricks. A human would actually understand the question for Please also understand the question and then answer it.
So this is basically the state of the art. Okay. Quote.
So again, some simple question is who is Michael Jordan?
Well, this actually very interesting for most people in the world when they say who is Michael Jordan?
You want to know the basketball player, but there's a machine learning researcher called Michael Jordan.
It's a pretty cool name to have the way because I'm sure that he's mistaken all the time for a basketball player.
Right? So only they're malgache people like you and me would actually want to know who R.E.M. Michael Jordan once recorded to get this right.
The new Artist. Edit I'm supposed to. When you ask who is Michael Jordan, maybe you can go by your press query.
It has set your whole search history.
And you asked so much about Al Williams and machine learning that it will then create a machine learning version of Michael Jordan.
If somebody has Google so many sports statistics, NBA stats all the time, then maybe I'll give them this.
Michael Jordan So interesting, right? Two different Michael Jordan's.
Who is Michael?
Jordan. Yeah. Ha ha.
The sports player won because, you know, most people in the world, right, do that.
So that is why Google actually went with that. But that's an ambiguity, right? So, you know, I'm going to say in MLA, then I can provide context.
Now suddenly, UC Berkeley cal professor you know is a Berkeley even here people ask this a different Michael Jordan okay but you and
I would not ignore this and actually go to the actual Michael Jordan Michael Jordan and Cal and read all these papers.
Okay. So I had to provide that extra words. That is called context.
I introduced context for disambiguate, you know,

and then it gives me that then so that I want so and all interesting and it's part of human would do as well.
If you go to anybody in L.A., it's always Michael Jordan.
They're going to tell you about the basketball player, but then you can go to some serious looking geek and say, who was Michael Jordan?
Well, you know, machine learning researcher. So it that the context, you know, there's no obvious answer because you know, either one.
Okay. So when this were when was Wendy's founded.
Okay. This is a pretty bad thing.
In other words, you should know by living in the world that when I lived in Columbus, Ohio, current Ohio State, outside Ohio, outside Columbus,
rather than going this wooded area and suddenly you see a mansion and then there's a little sign that says Wendy's founder Dave Thomas,
and that is where he lives. He's like a multibillionaire guy within the US from Columbus, Ohio.
So, you know, and this is a restaurant and that is what you're trying to ask game.
But if you purely go by keywords, then this is what you get.
Then you get this crazy.
The word sounded was in this document and the word Wendy's was written by the article was written by this person called Wendy.
It put these two together and that is the answer to when was Wendy's founded.
It's a pretty horribly, completely wrong answer. Right. You're looking at your grasping for words.
Yeah. Okay. That's what he should do.
That is why a lot of us have the challenge of knowing, oh, my God, Wendy's is a restaurant, so I shouldn't randomly pull words.
Okay, so when on the Ellen hallucinates live chat, you would you might actually do this,
but a Jeopardy that knows about restaurants and fast food industry wouldn't give me the right answer once again.
Yeah. For ritual augmentation over and over and over. That is the way to fix it.
Like don't try and do more like metrics and the weird crap, right?
Just just use drag records. Don't answer to the properly.
Okay, cool. So unless some more again one of us Microsoft established.
Okay, this could be a right answer. The word establish is not here at all.
Founded. Incorporated. Okay. But you know, this could be answered by some document that actually exactly had that sentence.
But then if a different document had descendants,
you would never see this document in the in the answer because that the word founded incorporated was not in the actual claim,
but with the knowledge graph of it in a lamp. This bottom thing can also be an answer.
Again, knowledge graph probably. Okay, I want to be. Yeah. Unleash the dragon.
Great. So these are all ways to all these slides.
Tell you asking a question automatically is not easy.
Okay. Because I let them just not live in the world. It has no idea about like when Microsoft were founded.
I was around when, you know, Amex was founded.
So I followed Microsoft from like day one, really the first PCs, you know, Windows DOS 1.0, all of it really up to now.
And so I would have a very different answer. I can tell you, I can just go on and on about all night long.
Steve Ballmer just on and off, but does not know any of that.
So you want to be careful. Look at this one. Where does the distance between, oh, this is so bad.
Okay. See, that is an actual calculation of some sort, Right.
In fact, if you had, like the coordinates of Sacramento and the various capital, you can find the distance and actually answered in minds.
Right. But if you're simply use words, okay, you're completely screwed.
Okay. No idea what it's going to say when you ask that is giving you an article about Nevada County in California, which is a completely wrong answer, like, Oh my God. So imagine if BART summarized it for you.
There is a danger so some kid would get it completely wrong.
Answer Okay. By the way, after you summarize something at the bottom, it says, Here are the links that the answers

came from.
Then if you look, you will know that answer is wrong, right? Many, many, many people increasingly will get used to not caring about the links.
Nobody cares about the facts. Okay, well, just tell me this somebody. I have no time.
Well, then you get wrong answers. Live with it. But direct can fix it as well.
So more and more. Right? You get it. Okay. And this one is like distance calculator, but only in California.
And this is this is from any city, any other city.
This would have the answer, but it does not still know what the largest cities they leave it up to it to figure it out.
They tell you, here's a distance from any city in any of the city, If you know what the largest city in an average Las Vegas,
then maybe you can then plug in Las Vegas, in California, Los Angeles, or look for themselves.
They let you do the work. Okay. But they can know or not let you do the work.
Meaning we can, you know, use rag for this as well.
Okay. I mean, I won't read all of them. Same thing. Put this in the mathematics in 1986 when you asked that it gives you like doctorate degrees.
Are they worth it? Like what the [INAUDIBLE] right is relevant? You know, there's some core question.
There's no way to ask you because it is going by keywords. So keywords are very, very bad.
It is hard to say. The opposite is called semantic search, meaning best search.
Understand what I asked you and you either know it or you don't know it.
But don't tell me like weird crap irrelevant. You know, this relevance metric, it fails irrelevant.
The top class irrelevant. Then who would have the patience to go look into all of them?
Just give up and say, you know, you screwed up in eight or No, I don't care.
I walk away. But someday something can actually find it for you.
Okay, so Siri, ask What's a knowledge graph in Alexa ordering?
What are they all do? Okay, so Siri does entity recognition like in a knowledge graph.
Okay, It's called named entity recognition.
I mentioned this a few times in the past, but I'm going to tell you again, it's called honor named entity recognition.
Yeah. Yeah. A named entity is something that's already popular in the world.
Already in the world. L.A., USC, Caltech, Rose Bowl, Venice Beach.
We all know what that means, right? You know, China, Oceans and, you know, Paris, Eiffel Tower.
So once you know all of those knowledge going forward on entry.
So if you say, you know where Stifle Tower is going, find Eiffel Tower in the knowledge graph,
there's a named entity, then it can start telling you what Paris did, you know.
There's also Notre-Dame Cathedral. There's also the river. All of that, right, can come from the knowledge graph.
So that is the best way to do it. That is what city does. Okay. Likewise.
Ask.com. It finds it, first of all, tries to classify the question, Is it a factual question?
Do I need like more data from different sources to calculate which is a pretty bad idea, by the way?
It is hard to classify questions into different types of game.
Once it finds the right type of question, then it'll use the right type to go in the index and then get the snippets.
It doesn't really actual documents. It only gets snippets from each relevant document and summarizes all the snippets for you.
That might not be the best idea, but just what Ask.com does. So Watson does, both of them combined.
And then Google also does the same entity stuff. But then now they use a knowledge graph that they built.
Yeah. So every company has knowledge graphs, which by the way, is a bad idea because all knowledge graph should be merged.
There's no Microsoft knowledge graph in many knowledge graph because it's all the same graph.
How many ways are there to say USC is university in Los Angeles?
Right? Dumb to reinvent the wheel, but they do it because they're to each other or something.
Someday will merge all of them. That'll be pretty cool. All right.
So, you know, just different ways to answer questions. We'll do a little bit more this about serial killer culture and what society, first of all,
takes your voice and converts that to text and then tries to understand what you said.
And semantic again knowledge graph. So tries to supposedly say things like, you know what movies are playing at 7

p.m. near my local movie theater.
And in those words like 7 p.m., movie theater, local, and it tries to find the answer for you.
It's pretty cool, right? They're all named entities. Okay. Yeah.
Then yeah, again, same thing then. What does it use to find the answer?
So many things. Geospatial map, database taxonomy, like ontology, cluster data, or even things like Yelp, you know, or even reservation services.
And so this is where, again, lamps can go to all of them and get data.
Al-Alam Rag again and again and again. And I'm going to go on pushing it because that is universal solution.
I make fun of it all the time. I think I was a stupid idea.
I'm a sucker. But I must admit I'm useful when I say they're stupid.
All I mean is they're not intelligent, really, like you and I are because they don't live in the world.
But we can help them by giving them a lot of good information.
Yeah. Then Syria would then combine all of this and give you like a speech back and an excellent family.
Well, you know, but Siri, Alexa, all of them, they're all due for a very big overhaul.
And overhaul would be based on Edelman. Amazon is already actually getting there.
Apple has not announced a very big l. I'm just still working on it. Know they always come last, but then they do a pretty good job.
And I'm more than anybody else. USC has an Alexa Center. I don't know if you know USC, Amazon, Alexa Center.
So this way you can actually be a part of all of this, the cardio argument.
So that's pretty neat. So how do you augment that?
Yeah, I mean in the center, but how do you augment Alexa, you know?
I'll just say Alexa alarm. Alexa alarm. That.
Preview Alexa alarm technology meaning initially, obviously when I actually came out, there was no way in [INAUDIBLE].
But now we're going to have a little memory. Alexis, let's chat. What's on your mind?
How did my soccer team do yesterday? Yesterday, Ole rain, 1120 against the Portland Thorns.
Can you give me a recap? Your team was steady all through the first half with a strong defense and 52% ball possession.
Things got heated during the second half with a foul that earned the Portland third.
Not so much about that.
And it gets all of that from a text newspaper report where somebody just simply objectively stated the game even more amazing than this.
If you input a video of people playing like soccer, imagine that giving you a running commentary of the game.
Oh my God. Right. I'm not sure if I can find it breaking it to you.
I'll just say a malayalam video.
Soccer game. Commentary. It's a brand new thing, like two days old or something, and I'm not sure.
So it might come up. It may not come up. Okay.
Match analysis, gentlemen. It might actually be this one, you know, but what I'm showing you is actually much newer.
Maybe not. And I'll just quickly look if number one. Yes.
It's just crazy what you can do. See that? All aspects of it, you know, you'll actually do a commentary for you.
Airbus video game coming to an end just to this one. But that's all the.
Yes or no football players? Yes. It's all about this whole YOLO, right?
You're almost like a, you know, a vision based like machine learning model.
It segments the players so that it can follow one player and know like what?
You know, kicking the ball, for example, towards the goal. It'll actually raise it towards the goal, you know.
Okay. So. Hmm. I'll just say. I'll just say see?
See what happens. Let's see. You need more context.
Okay. Hmm.
Just say hello. Commentary.
Commentary. Automatic commentary. Commentary.
Hmm. Oh, well. Okay. I'm going to find the link compared to afterwards.
Anyway, so the whole idea is context once again. So if you give it enough context, in other words, to be so specific, then it's better.

Like a human being. Okay. So in this case, the elephant is a pretty cool restaurant.
There's one president now, one in Palo Alto. I live one street of the one in Palo Alto.
So then this animal restaurant seven with my mom. That's a little ambiguity.
So then seven is actually 7 p.m. That is a decision it has to make.
Okay, that's pretty cool. Yeah. And then send her an email reminder.
That's pretty neat. Her you know, you said this first of all, then they said her.
Okay, this one more alum comes in. I'm going to start you. Google has something called Bird an apple.
Re it implemented? Bertrand So Bird is a transformer. It's called bidirectional transformer.
It's to be inverter. So transformers are how the air can keep track of the sentence.
And it said the word mom on the next sentence and send her.
Then it'll know it's mom. Suppose you say something like book a table at L for an hour at seven and talk to my favorite.
I don't know, Frank. You know the maitre d and then ask her when we should show up with my mom.
And then I ask her when she show up at my mom and then make a reminder, make this thing with my mom and send her a reminder.
So then I have two hers.
On the one hand, I said, make a table and talk to the chef and say, Ask her when I can come up with my mom and then send her a reminder.
Now, suddenly this and I'm ambiguity. Should I send the maitre d that the receiving person reception is a text, or should I send my mom a text?
That's a real ambiguity. Humans would know because you're in your head. I'm taking my mom to dinner.
Dumb. And Mommy's not doing that. But then the. I cannot imagine.
There's no mental image that's not being formed. So that's the context.
The game. The context is basically the magic. So Transformers are really all about context.
There's something called a tension mechanism. Okay? And that is the pure magic inside Transformers.
I'll tell you another time saved by the action. That tension is actually from both sides by directional trails.
Like, okay, so that is so they can answer these questions about her being mom.
In other words, here is not a big deal. There's only one her the other people here that's ahead of her.
Other people like, you know, I'll told you, my friend. Then suddenly the her is not a unique thing anymore.
But this is pretty unique here. Okay, so Chicago pizza uses a Chicago pizza.
Then you have to ask, So what do you actually mean? Do you want deep dish Chicago style pizza or do you want pizza restaurant in Chicago?
You know, maybe are traveling to Chicago if it knows that. And then give your restaurants in Chicago.
But if you're not traveling, I'll give you a deep dish pick of the Chicago style pizza restaurant.
All right. Again, context, context. Context is hard for humans also.
Then it's all about scoring, you know, and basically just says, look, heartless.
IBM, Watson, Oracle Khan said, generates many possible answers.
Look at that. More than 50 components, the most from 50 different knowledge sources.
Okay. And find some bench answers and then you score them and then find the answer at the best score.
So we don't have to read all of this. You guys look, just, you know.
The US, for example. Like what? Source? In other words, what is the authority that all this signal looking like was a spacing,
like where you're asking the question from when it's a question about and also some kind of taxonomy ontology.
So many different ways of combining pieces to answer your question.
And this all three, remember, IBM, by the way, has had this thing called Watson.
They call it cognitive error, but they're still pre chargeability.
They're all projected such a pretty fast. Jeeves, I told you, you know, they do.
Basically, the take from the question takes segment some keywords and then try and find documents
that match those keywords and then use snippets from them and then formulate your answer.
So it might not be all that accurate in on some people. It assumes all the time that so far weaker than, you know, such as Google.
It used to be great until Google came along and I was like old, old fashioned technology.
No point in talk about it. So named entities.
These are all named entities. Okay, So people are named entities.

Pam is also named in the Second World War. And I you know, like when in 1945.
So location is always named in what, again, this small location.
And then sometimes you in numbers, famous numbers. You know how many monsters Earth have won so the numbers become important.
Okay, great. Then again, how do you answer it?
First of all, have to process the question. Okay. Destroy the whole named in a data condition comes in, you know, so, you know, six W plus H, right.
When a reporter an article about something this the formula that they use.
Where does this mean? Or maybe when fired, if you're only one of them.
Okay, so let's do this simpler. So what is five W plus H?
You're writing a factual article about something. What would you say?
What questions do you ask? These are all words. Who?
Who? What about where?
When. And why?
Oh, and this is how.
Yeah. If you answer all those questions in the article.
The whole article. You know, actually if you now five W there's also six W Right away.
Okay, cool. Anyway, so that's why this is so I get that and then go and get actual documents from your PFI collection, but not to document just snippets of snippets because it's shorter and easier.
Then you are break down snippets into little pieces and then try and match them with all of this and see.
Then use and combine all the snippets answers and then rank them and then present them.
Wow. So it's all coming from snippets. Okay. Not doing actual documents because the actual document might be a 500 page PDF.
He's not going to read through all of them to answer the question. And that's a very fast train.
So they use a summary which is never quote.
Okay, then once again in detail, but just exactly what we just said.
So you ask a question and then you go in the document, but get the snippet passages and then answer the question.
You know, so again, we're leaving out some of the details, but, you know, this way,
the whole any unnamed entity recognition word in the taxonomy hierarchy.
Okay. And then possibly think process can Google.
It is not the precedent I have in mind. Okay. Okay. Oh, yeah.
This a bad idea? Okay, so, you know, at one point, Google tried to actually classify questions.
Okay? I mean, there's so many types of questions in the world. Can I just classify them very broadly?
You can say that's a factual question.
Like I asked you, what is the population of the world? Approximately 8 billion.
But like asking more complex questions like should I pursue a career in data science?
But is not a factual question. There's no fact right matching you more like what is your opinion?
So loosely speaking, questions are like that Objective versus subjective objective means there's only one answer.
So it means interdependence, right? But other than that, all bets are off.
You cannot classify questions into the question about a location, question about a product.
What the [INAUDIBLE]? Right? Mammal. Reptile. What about birds? No question about a game.
Go on. What about sports? So it's a it's full of holes. So that's a crazy thing.
Question taxonomy. So don't even go there. Just give up. Okay. Yeah.
So this question taxonomy, so like, so crazy. So possibly says, why does the vaccine prevent.
Yeah, that's a question about disease. Okay. They're trying to classify actual questions into some kind of a high level category.
But the problem is there are way too many questions, way too many categories. But actually I feel great.
So that's that's all that started. It didn't work. Okay.
Yeah. Who was confused? Yes. You know, I mean, look, a Chinese philosopher, all right?
But then it says human, you know? Yeah. I mean, on one level, that's right.
But maybe that's useless. As a Confucius human, right? You should say he's a philosopher, right?
It's a better answer to sometimes entity classification can throw you off, you know, It's all right.
So again, how long was Maoist March? You know, that's a number.

Obviously, there's almost like a function call. Okay, so what is the input?
What is the output type? You know, trying to classify that? The answer is sometimes not that easy at all.
Well, that's is factual. Like, you know, for this one what is us is, you know, main like number.
I can Google that and come and actually tell you two and three you two to the way what what right that's easy.
But the more complex questions are, the more vague questions or comparison questions.
No way you can classify them. That part fails. Okay.
Yeah. Then you know, this this whole the whole NLRB stuff, right?
So I ask the whole question. You go to the more like, pieces of the question and only give the main keywords.
Right. Stemming limitation, all of that. Okay. And then likewise the any are so in the question actually see if there are some
pieces that are pretty famous and then go and knowledge graph and then the
semantic relations in the words know like you know when they say from London
to Los Angeles that's a different thing compared to Los Angeles to London.
I understand the question properly and I'll answer the semantic relation.
And then likewise, you know, you can use a similar word concept dictionaries,
other words, you can augment your your answer by looking for the same words here,
actually the same words, but also analogous similar words in a Tetris, like I might say, sad.
Then you might say something else, you know, a different word for said eternal and dull, depressed, and that'll be a better answer.
So do all of these. So all of these are basically loosely called an LP.
Okay, cool. Yeah. Now you see the NPR.
So in this paragraph. TRUMP You know.
SCHMIDT FBI, Hillary Clinton, Lisa Page You know, Robert Mueller, Trump struck, struck, struck Strzok.
Okay. And then the FBI, Peter, again, these are famous people, right?
So Peter struggles to work for the FBI. You know what? FBI is an entity.
So once you know, from some big passage, some famous names,
then it go in the knowledge graph and augment every single one of those names with extra things.
You know, it's all. Great. So you recognize known entities like here in Washington know instantly.
We know that. And then now your training sets in on this part.
Oh, yeah. One less thing. Actually, you can use an alarm to even build these kinds of knowledge graphs to actually do any are.
Okay if I give you a big file. Okay. And I said are you going to extract famous any yours from there like extract all the famous places you know
names that used to be a difficult task because somebody has know what is famous and what's not famous.
Right. But now the L I'm showing the basal alarms are trained on so much that we can use and l am
actually taking on structure document to construct and they are using l m to construct any eyes.
Again, please do these kinds of tutorials. You know, there's python modules that you can use, go tutorial,
some random passage and you will actually see it's amazing what you are able to do with them.
So this way humans don't have to be involved, can basically give it unstructured, crazy text and say you made any of it because it knows that.
Wow. Okay. So then that's what that is. I can even give you a tutorial if you look forward.
Right. So graphical towards error. Yeah. Some of them, like I said, are more detailed than others you can read.
Well, tangent there. Yes. So then ultimately what happens?
Right. So when you extract from a document all these any are keywords, you put them in a knowledge graph as a whole point.
Say there's John Lennon and Yoko Ono. And you would kind of divide between them and say, married know.
You can go from one to the other. And then all the animals that you pulled out will all become part of this knowledge graph.
And that can go on getting bigger and bigger. So that's it.
So in other words, take that sentence, get any ass, and then from there, plug them into a knowledge graph.
Quote. Okay. Keyword selection.
Yeah. This all about questions. Okay. So in this whole question, like, you know,
how many songs that the Beatles sing and you don't need words like in how many or how or something like this one

songs Beatles, you know.
Then they pull out the keywords, right? That's what all of this is.
So identify what the noun singer's like, a verb, for instance, and the non stop words.
Yeah. So some words you should not throw it in aren't actually stop words.
Okay. In quotations, every single word matters don't write out.
So these are all things that the search engine will do before it actually answers your question.
Okay. And so type again the answers belonging to different types, I guess.
So pick the specific answer. Type like a subject to answer, subject to answer, that kind of thing.
We'll take a break, right? Okay. You'll break and then finish the rest. So then again, how do you expand a bunch of keywords?
And a question can expand them in many ways, right? One of them is if the question was, you know, who invented the telephone?
And suppose you have a page that says the inventor of the telephone is Alexander Graham Bell, the word inventor and inventor to go together.
The inventor invents things, right? That way you could take one word and then convert that to other words and still search for all those words.
That is expanding one word and two more words. That is called a morphological variant.
Morphology means form for the words form like present tense can be expanded to patterns, you know, or singular can become plural.
Just simply a word expansion, not like a tree taxonomy expansion.
I'm not saying inventor is a person not doing that, okay? Just simply going from floating around.
We can do that. Or you can use that SARS and look at synonyms.
Killer, assassin, murderer. You know, they're all the same, right?
So even though I search for one word, you can search for other words that are similar and then pull out more answers for me.
Likewise, for how far is Los Angeles? You know, from from San Francisco.
Then the document says this is between San Francisco and Los Angeles, 500 miles.
Then I should be able to use that document because far distant, they're related.
Right. So, you know, use things like that. Or again, this and this are like almost the same lexical variant.
I mean, it's just similar to synonyms. These are also like synonyms.
I mean, and then and then these are the same like prefer choose, you know, pick God.
I mean, the same thing for you and I type like, you know, the cool electronics, you know, then you're sort of prefer choose who would choose.
Join us for breakfast. Why not a pick so I can use variance.
So these are different ways to take the word that is in the question and make more
words out of it and expand my search and bring a better answer back to the user.
Great. Okay, so we're in it. So when it is a little bit different at all, you you go up the hierarchy.
So somebody says, you know, who wrote a particular poem recorder stopping by the woods on a snowy evening and said Robert Frost.
But now Frost is also a part and poet is like a content creator and a kind of artist human being.
Then I can use all of those extra things that I know about Robert Frost to answer that question.
Then I go up the hierarchy, okay, And then the word net can be used for that because the word net, all these hierarchies already exist.
Once you know somebody is a person, well, once you have somebody support point network support is a kind of writer because you write poetry and
then the writers communicate a communicator as a content creator or maybe a creative human being,
an artistic person, you know, by the way. Although we are yet to.
She writes poetry. And she paints.
Good afternoon, Professor De. First. Dan.
Good afternoon, everyone. I am Jeff being well. I'm glad to become a student of Professor Tang.
I come from the team of be I Jeff who I and shall I.
I've been addicted to literature and art since I was born.
The high scientists of Challis not only gave me my appearance and my voice, but also taught me to compose the

background.
Music to hear is composed by me. What's more, based on the large model Dell 2.0.
I can also write poems and paint. However, I still feel it's not enough.
So I came to the university to further my studies.
Singhal A university has a prestigious reputation for computer science.
I became interested in my own story. So I look pretty amazing.
There's also, I think, look, I think it's going on because even more large level language models compared all of this, Alibaba has one, basically bad companies, all of them. How can this in a holder on really this amazing some of them are bigger than chargeability for.
Okay yeah the Chinese I don't know if there's a whole bunch of them they are amazing.
So what's this? The Chinese alums. Bigger than jets are pretty bigger than Djibouti for.
Yeah. Ernie, without really knowing what entry point out on this many more.
Okay, so we have no time for all this. You can go read it. I mean, this is. This is nuts.
See that? Yeah. I mean, it's a thought is 1 trillion parameters, but parameter.
We simply mean for all the neurons. How many inputs around. So the bigger it is and obviously the bigger the model is.
So this a lot here, right? Yeah. Channeling.
Uh, so many. Yeah. Ernie.
And then. Now. Oh. So minute that.
Yeah, See, I mean, that's crazy, right? This is this again.
The race to get look who is going to become bigger. Right. But at the same time, there's also a different race to say who is going to get smaller.
That is the whole Mistral vicuna. Llama already told you.
They're both equally useful looking. All right. So then, you know, this is about the whole entity stuff.
All right, So then once you find a taxonomy, then you plug your actual words into the taxonomy.
Meaning, if Ginsberg was a poet, you can say things that you might say about communicators.
You might say, Ginsberg You know, it's pretty, pretty good at giving public speeches.
You can answer it like that. Then that's like intersecting a big taxonomy.
But just this little part that you found in the taxonomy and just using that part of the taxonomy and then a section.
Okay, So right here. This has merged, but some can in a section, meaning all of these people are actually a person.
So in that sense, you don't have to worry about like all this, that a person is not a date.
Right? Or they are not money. So they only belong in this part of the hierarchy, you know.
Okay. This is the part where you have no entity. Some words from from your keywords.
Search, expand the words and make more works.
But now you need to go to your TFIDF and get documents, specifically snippets of documents, and then put them together to answer a question.
That's what this is about. Okay. Like this. Get the snippets and then rank passages.
You know these things we've seen before. Passage means leopard passages.
Okay. And then in this network, you look for all of these. Look for nurse.
Look for keywords, You know, anagrams. All this coming from snippets.
And basically then rank all the snippet results somehow and actually merge them.
I think it might say this in a similar way. You have to merge the snippet results.
Okay. Doesn't say that here. Then you get one coherent like English sentence.
Really? Okay. The last part is this.
For a Firebird slides. Okay. Okay. So in about 2017, so much revolutionary things happen.
Now it's only 2020 and they all happened so fast. Especially hard to keep track of them.
They're all happening like a little clump of time. What happened was Google made it one paper called Attention is All You Need.
That is pretty much what really started the entire thing off. Attention is all you need.
That paper was like a watershed moment in the world.
Before that paper world. After the paper, the paper change the word forever, literally.
Until then, I was hopping along like some narrow success here and there.
Right. But this wasn't like, Oh, my God, you know, how did they come up with that?
So then the idea was that I'm going to tell you that now, but to people, right?

So Vaswani and also Nikki Parmar, the board.
Ah, from Rotary. Okay. But you are not from here.
They work for ISI. They've got to do that. Very cool.
So that that paper is basically. It is magical. Will go to some of the time, but it's also a hack.
Okay. The papers look all kinds of cool little hacks in here.
For example, at one point is divide by square root of a number of tokens and divide by square.
It has to take care of vanishing gradients. Okay. Okay. So it is not particularly brain like or anything.
Okay, so instead of numerical computation paper, but it is nothing short of magical because our brain works in a very, very different way, but it generates a sentence as we speak. This also generates sentences, and it's not a brain, but the sentences that it generates.
If you forget the hallucination, it's like the words themselves are not there.
Perfect grammar and long passages can write and tell stories.
Okay, like what the heck want from that? So then they went to Google actually, sorry, the word Google when they publish this story.
Then. Then they went to open. Actually these people went open air and now they actually quit open air and went to the competing company.
Okay. I think there are entropic probably even luckier than Google.
Google kept going and they made this thing called a bidirectional transformer.
So Bert so Bert is now the next thing that they made and then they made even more.
They made one called Ernie. They made one called Roberta.
Roberta is based on Bert. There are so many of them. Okay? I mean, it's all hierarchy of them.
Okay, So then after that, after you read the first paper, please leave the second paper.
See, all the authors are different bunch of people. And also a little bit later, like one year later.
So Bert has a bidirectional, basically context mechanism.
It's all about context, I can tell you. So the thing that made Transformers pretty amazing is this.
Before Transformers you had a long sentence, which is called a sequence learning task.
I want to know what the words mean, that one word after another word.
But the problem was in a recurrent neural network, or at least long short term memory.
Those are the two competing architectures. They're kind of similar, but a little bit different.
We used to follow sentences like this.
There are two existing strategies for applying Pre-trained language representation doesn't ask,
and they say how many existing strategies hopefully will set up because you are remembering the words that came before something, right?
That is called maintaining context so humans can maintain long context.
You can basically imagine Harry Potter in you had the last chapter of Harry Potter can go back to the first chapter and ask a question.
You can answer it. But Alan absolutely could not do it in the past or an intellectual moving train.
The train is the train that's five carriages, right? You drag the train through these words.
It remembers those words. You can look forward. Not that, but as soon as you drag the train further, you can hear these five words.
It forgot all of this. You it only remember like a tiny context moving window.
Okay, that's a pretty big problem. Like part the horrible short term memory.
Okay. There's absolutely no long term memory about failure.
Listing velocity group transformer. Its context kept increasing based on from the first word, almost all the up to the last word.
Our long and confident memory, the context. It basically never forgot anything that came before it.
Okay, that's the best way to explain it. And I was pretty crazy. But it's only in one direction.
From left to right. Bert does the same thing backwards and forwards.
It remembers context backwards and forwards again, and that gave rise to like you and better performance.
And this is what they use for Q&A. So to come back to all of this. Okay, so today's Q&A and ask a question, use Bert should answer a question.
I show some slides. And all. Should we do a break or not?
It's up to you. Okay, so all of this, right, does look what it is.

And Bert has a page you can visit. Like this award is produced for Q&A.
In fact, Transformers back then in Google was invented to actually do language translation.
Shocking. If you go in the paper I showed you. Attention is all you need.
Paper. The how in English, the French translation. Now that is what I thought it was good for.
But open air, you know Ilya and people like that realized, Oh my God, this is much, much bigger than just language translation.
We can actually create new words. You can make it generative. That is when the religion actually was born.
Okay, So Q&A was obviously a bird can do that.
You can also summarize like a big passage into a bunch of smaller words, abstract summarization, and then it can do sentence prediction.
So this is called mask training. When you mask some words and then you train the neural network and basically guessing what those words are.
Okay? And they gradually predict like words, sentences. And then, yeah, so this is literally what it is.
You can talk to it and then generate a response. Okay.
So you can do all of that. And then so it can also be useful.
All of these extra NLP task is a classic interpretation. I mean, that is really what you run the A.P. course, I give you your tweet,
you're telling me positive sentiment, negative sentiment, you know, happy, sad.
So there's all this bird paper you can read, okay? You can just do it. In fact, I sort of do that.
That's the bird paper. Yeah. And then these papers are highly readable, by the way.
Yeah. You know, just not too much, Matt. They just completely like how it works.
Spend some time. I'm going to find out my sources from, like, a LinkedIn medium dot com.
I'll put together a nice collection of transform articles and burn articles, many with a video menu within a notebook screenplay.
That is the best way to learn and not read these papers. These papers are hard to read because they're very dense, you know.
They don't explain anything. Just simply tell you what it doesn't need to know.
Okay, so then Burke has all these so-called pre-trained models, in other words.
So Bird is the school idea, right? Then Google and others too.
But. And made so many neural networks, you know, learn.
In other words, this one one year network second, one third, one fourth and fifth time.
They're all trained on different data, and so they call them pre-trained models.
The word pre-trained comes from the following. When you train something, say you train.
We say, See, train this one video back on all about videos of a YouTube videos that is not called Pre-training because you're doing the training.
Okay, but the word pre occurs when I take this train model and I give it to somebody over here and say, you know,
you can build this whole video chatting application, knows all about video then for that person that is already pre trained.
Okay. That's what the word pre-training. So predictions for general generator pre-trained transformer.
It is pre-trained not very open. An open air train that is pre trained for you and me when we can use it like an
alarm and make trouble with a generator is when it actually generates words.
Okay. So that's what you really sense with fine tuning.
I told you to take something that's already pre trained as many pretend models and take any of them and adapt them to something new and more specific.
You know, for example, in all of this domain, specific cannot be.
It can be about doctors, can be about lawyers, can be about code, can be anything, really traffic.
You know, when you get something more specific, call it fine tuning.
The alternative is don't do a fine tuning limit as it is. Go externally over here and do it to L'augmentation once again, two ways.
Okay. You can you can find out the pros and cons of both of them. Fine tuning versus interact.
The problem with fine tuning is you need to have access to the actual level.
You cannot you and I cannot fine tune open the activity for it because open air holds on to it.
Right. As knowing how long and let me change it so you and I can only drag on it.
But if you have an open source lamb like vicuna or llama, there's so many Mistral, you have the entire source for the

alarm.
That is when you can find your name. So, yeah, people tell you all versus, you know, I'm in all of this, right?
Which is the best tool is really literally there's so much. But in the end, they both do the same thing.
They become more smart than just a generic and overall alarm.
Alright, so that's that's all this is, you know? Okay. And then Google would use the fine tuning once they're doing general search.
But a company that is not Google, they have a patent database.
They want to search patterns all the time. They're confined to in their element patterns and then you start to, you know, answer.
Okay. So then the whole bidirectional, you know, context of it.
Yeah. So reading, but it's almost like you read the sentence this way.
The animal did not cross the street. That's so far so good because it was too wide.
Really backwards. Wide too. Was it? Because it's a loosely.
But the question is, what does it reflect? Okay. There are two nouns here.
Animal and street. What was the word? Was it animal to word?
Or was it street to wide to restrict where that is?
What context use it. Okay, so the whole bi directional business, but I'll explain much better if you just wait.
So not now. Just just know that the context is being computed basically from both directions.
Okay.
By the way, when you keep track of all the words that came before you and then you know that it is animal that is also at attention means the system.
The whole transformer is paying attention to all the words that came before it.
Even the new words are coming after you pay attention. That's why it says attention is all you have to then that.
Yeah. Okay, look at this one. So you ask the question, right?
So I go in Google and type what? Where the water droplets collide with ice crystals scale.
That's my question. Then the say the question was says to the corpus, and this answer came up, okay.
Now, how would you use bird to say within a cloud? Because see, you have all these extra words.
So you should know based on this question that these are the only words that matter in this whole paragraph.
So you don't need like all of this to answer that when you look at it, right. Where do water droplets credit where And that's what be within a cloud.
That's a place, Right. That is where the answer would be within a cloud. So a would be used to basically delete like all of these.
Okay. Because it's the only thing that matters. So it's like a summary of what it does.
All right. So starting in tokens, in other words, this a token start out as a token.
So like where within the start of my answer that one should have Stop, stop right there.
Wow. Then you get a sensor. Okay.
So I just simply anticlimactic. That stopped right there. You know, what else should we do?
I mean, it's eight or nine. So you're telling me. I'm just telling you so much, right?
If you're it heats up and you say, Well, watch this stuff. Or I can tell you a tiny bit more about the last part.
You know, the third thing that I had in mind, I'm mainly behind you guys,
but not worried because, you know, we can catch up, but I don't want to be too much behind.
That's all this way, because I want to, you know, have some neat things to tell you here and here.
I want to not, like, dump everything on the air. Okay. This is an easy idea.
I want tell you the whole thing. I'll leave you alone, okay? But I'll tell you the basic idea.
Just like in the real world, people are clustered by their own count as distinct groups of people.
Again, this goes back to the whole embedding idea I should actually show you.
Okay, I'll just say that it's about clustering as well. I'll just say vector DB, I'll just say semantic similarity.
I'll use it to explain clustering to you in the real world because of sentences used for describing very specific things, organic chemistry, right?
How words like life molecules, precipitation biology, you know, I don't know.
And likewise something about crime where how things like murder, blood murder weapon, gun, you know muttering I don't know Right.
Law enforcement.

So therefore, based on how we use English, write different words, different sentences end up at different points in a multidimensional space.
I mean, look at this one. This is basically what I'm telling you. So all the words about banana.
Okay, bananas, quite long. Bananas, yellow bananas. Good for your bananas, potassium, you know, vitamins, all that.
It'll all end up in this part of the space, so to speak.
All the words about chicken, all the sentences about chicken wouldn't appear to naturally cluster.
This one does not look like it's cluster of all these cluster, all the animal stuff in the here cluster,
all the fruits, cluster, all chemistry, cholesterol, computer science cluster, all things about geography.
In some kind of semantic multidimensional space, things do become cluster.
Then it's like a clustering algorithm for data mining. For each cluster you can find a cluster centroid cluster centroid, that cluster centroid,
that cluster centroid, this cluster centroid can be called animals.
Okay? The centroid, if a new document, a new word, a new query ends up here, then you would know.
I should classify that query. The document, that new thing that came in as related to animals.
Wow. If I. If I knew where your new book came in and it ended up here, I would know instantly.
Wow. That's a book about, you know, fruits or it's in your query about foods.
So clusters happened because real world sentences, real world words use different terminology for describing different things.
So as long as it's a border, that's a gap between individual clumped entities, you can draw a line between them.
That is classically the idea of clustering. I already told you this earlier in a previous slide previous lecture, but now you can go.
But it's about this situate you and then it all comes down to.
Comes down to. Okay. You read that?
Yeah. So supposing you have that as your input with all the documents, I would naturally say, you know, there are two clusters.
Why? Well, because I can group these as one unit, clearly.
Right. You would not say that requires a you only say two. But within this cluster, somebody can say, well, can you make more clusters?
Say, this is physics, chemistry within chemistry, physical chemistry, organic chemistry.
You know, like in organic chemistry. Why? Well, because I can see some gaps between them.
As long as you can see gaps in a multidimensional space. Clustering can happen.
And once a cluster, you can classify because classification means again, a new query comes in here.
I'll only use this answer, make a new document and it appear I would call it what do I call all these documents?
So cluster first and classify is a thing. So that's ultimately what I was going to tell you, like next week.
So the same can range clustering algorithm is really what is used here in multidimensional space.
Okay. You know this from data mining and for clustering, you try to find distances, right?
I told you, you actually calculate like what is far away and what is nearby.
There are so many distance measures. Okay. And that's also very interesting.
And to go backwards, you're going to know this probably from machine learning.
Supposed to say this, A cluster is a cluster. Somebody would say, Why?
Well, because any two points here are close compared at any point over here.
Conversely, any two points here, the closer compared to any point over here, this large gap between them.
So when cluster. What do you mean gap? Well,
Euclidean distance I can literally measure in three square zero squared minus square x1 square X0 is excellence squared plus
y zero minus y one squared x0 x mean I can use actually equal in distance or I can use what is called Manhattan distance,
which is I simply take the mark of this difference.
I'll do x zero minus x one between these and do a modest opposite or negative disappear.
Likewise, do a margin in the y axis and only do distances based on x and y axis that is called Manhattan distance.
Okay. Well, I can even do cosine distance. I can convert it into a vector.
I can say that's one vector as the second vector small angle between them compared to that train.
That vector big angle cos of a big angle is basically going to zero cost 90 zero zero is one.

So I can use an angle as a distance measure that all kinds of distance measures and we can use any of them to cluster.
That's the whole idea. So we can come back to all of that and definitely look at it again.
Just three random things, okay. Oh, the clear Euclidean distance.
So this two L1 norm, that is the Manhattan distance. Actually, that's a different one.
Yeah. So this one, so L1 L2, this one L2.
You probably know that, right? Because L2 squared L1 means linear summation.
There's absolutely no black power after all cosine similarity. So I can use these norms.
There are many, many more distances, by the way. Okay.
Make sure that you know and again, stop with that ML distance metrics and use any of them really which I apply or typo.
Fix it. Okay. See that? So Euclidean, that's what I told you is called taxicab geometry.
Only measure along X and Y. Okay.
And then you can do Minkowski distance this a lot like Euclidean, but widest squared gotten into three or even like 0.5.
Now I have a curve space. Okay, I don't know, flat space, Euclidean flat space.
You also have, you know, hyperbolic space and also space dormitory.
That is what that is. And of course and distances you having this and now this based on bit differences so as a card distance right Mahalanobis
is very similar to Manhattan distance but with a power and even you can use these are distances L distances.
There are so many distance measures. Okay. But in all of them for clustering, we only use Euclidean mostly, but we can use Manhattan possibly.
And then of course. And possibly. All right. So that is, I think what we'll do next time.
Okay, So next time we'll do that and then just do a little bit more.
I truly want to tell you the new study, one more so know. I hope I can get to that.
All right, then. Remember the homework submitted and then you're going to have fun with the last one.
Well, I take this out completely, so no more of that. But instead, I like to part B.

Lecture - 3b

Hey, what happened to everybody else? This time I only got 30 emails from people that wanted to be away.

That's like way more than 30 missing. And I will do attendance for sure every single time.

Right. It. Oh.

Hi. Cool.

Hey, let's start. All right.

The first thing I want to say is today after class.

I do have a meeting with someone in China, so I have to rush.

So one have office hours. Okay, so to speak. Office are informal office hours.

But then, if you want, you can obviously email me or post on pager and then personally shout out to John soon.

Oh, I got your link. Perfect. Yeah.

Okay, so what else can I tell you? Yes. Oh. Homework for.

So I'm actually still working on the second half of your homework for. I'm trying to make it simpler.

Okay. Okay. So homework for part one. I already have.

I could give it to you, but I wanted to give it to both of them. To you together 100% by Friday night, which is tomorrow night.

Expect both. Okay? Please look for it. But not tonight.

Homework. Part two. Homework for part two is going to be a little simple homework,

but this time you will do everything locally with a little larger language model and then you will download it to your machine.

It's about ten gigabytes and then use Python to actually so-called chat with it.

In other words, it will ask queries to a PDF file, and the PDF file is also local,

and then it will hopefully answer questions from sections and that PDF file starts a new form of retrieval.

Right. So I wanted to explain, I wanted you guys to experience that.

Otherwise this class is very interesting. I hope you learned like so many things about how things have been done for so long.

Today I'm going to show you clustering and classification. Clustering and classification, especially classification.

It leads you right up the chargeability believe it or not, it leads you right up to something pretty powerful called Attention Mechanism.

So now architecture and your network architecture called Transformer, that is something called self attention.

But before self attention, there's a previous mechanism called attention.

Okay, So that is what this whole classification stuff is.

It's so close to the whole revolution that is happening in the world today.

That means after we come back from Thanksgiving, I want to tell you, I purposely call this assorted topics living entirely like blank,

meaning I have a whole bunch of things that can all be very short, 10 minutes, 15 minutes each.

But I still want to tell you a lot, starting from what is it transforming you?

How does it work? So that is really the basis of everything. So then from Transformers could charge a pretty well.

By the way, even there, the whole charge of beauty uses transformers, but is more than a transformer.

It uses something called human feedback. And guess what? In this classification lecture, I'll talk about something called the rocket algorithm.

The rocket algorithm is human feedback. So these things are just amazingly related together, you know, these ideas.

So anyway, in here, I'll tell you about Chargeability, which is Human Feedback Incorporated.

And then I want to tell you about this thing called land chain. So long Chain is a prime programing language.

How do you augment, you know, how do you basically make problems automatically know how do you maybe like expand your product like lots of things?

Okay, How do you have like a template for a prompt and fill the template with actual words and make the problem bigger?

But that leads to the next idea called agents are functions.

Actually, you know, how do you make like long chain functions?

And then gradually from that something called an agent, which is really, you know, a version of chargeability,

meaning one instance of LGBT running off and doing its own thing, like writing code.

But how can you write many agents, you know, and then make them all interact with each other?

So it's getting more and more powerful. So I want to tell you about that.
I also want to tell you about the whole notion of a large language model of the word GP.
Default is one of them. You're pretty famous, already rumored to be in the works, but it's also small language models.
So I tell you why we wouldn't care. I mean, I have so much to tell you.
It's not even funny, but it's colder than the whole asylum, small language model,
all of that and luncheon leads to a pretty cool idea called retrieval augmentation.
So that is the only way to do that. You're going to get charge a pretty to answer truthfully, whether it's scientific,
whether it's law related or about your bank or about products at droughts, there's truth in the world.
So when you ask a search question, you hopefully want the truth, not just some words that look like the truth,
because I'll just simply compute words that look like the truth.
So it has no way to go back and verify. Okay. So that's really very scary.
The core idea, which by the way, is not even one year old on November 30th,
exactly two weeks from now is when it'll be the first anniversary of Chargeability.
So it didn't One year the world has completely changed. Look, there's so much innovation, people piled on it, you know.
Meanwhile, the image generation had been going on for two years before that, you know, since they're in the pandemic image generation.
So together, all these things are now called generator because they're able to generate words to charge repeated generating words.
But then there's a very different mechanism for generating pixels in video.
You know, that's like diffusion algorithms, for example. So I wanted to basically tell you so much in the sort of lecture.
Nothing would be like very deep because we have no time.
But I want to point out like basically 20, 23 bleeding edge and then you can take it from there.
Okay, I. I want to point out things like what is a context window?
In other words, how much can change a predictor, remember of your past conversation, so to speak.
You know, so that is limited, somewhat limited materials and techniques to go many, many orders of magnitude beyond, you know, jeopardy.
Oh, that's crazy. You know, that that's happening. So in other words,
this thing called large language model in our selectivity is being pulled in so many directions by the world because it is such a great idea.
I in retrospect, you know, in hindsight, what we had until then was cool.
But it all looks like child's play. You know, even the convolutional neural networks that are all cool are a supervised machine learning.
But suddenly they all seem like a joke, almost like, wow, what the heck, You know, so unintelligible, all of them.
And then, needless to say, if it is that powerful, it is going to also involve all kinds of aspects of,
you know, ethics and bias, fairness and or legality, many, many things.
So I'll point out those as well. You need to learn them together and then we'll do a little review.
Okay. Like one hour each approximately, or maybe one and a half hours and the rest, you know, this.
And then to remind you, next Thursday, exactly a week from today, you know, Thanksgiving.
So absolutely no class at all, no zoom, no ritual, nothing.
So just stay home, okay? Yeah. So that is why today's these three topics.
And then is that so? Then I'm going to basically just click on clustering and get going.
Kind of like a lot to tell. You think I told you, but yeah.
Okay. Yeah. Yes.
So let's do it. This is wonderful. Also, this clustering idea of clustering documents.
It turns out clustering web pages, you know, is related to Jeopardy!
It's called embedding. Okay, So clusters are like, you know, clusters of what are called embedded points in some multidimensional space.
So in terms of image search retrieval, rather, so what this what clustering means,
you have a bunch of documents, you have a bunch of web pages that could be in certain topics.
So this is more than just a keyword search. You know, that these are physics documents.
You know, these might be astronomy documents, these might be biochemistry documents.
Then the idea would be when the user searches, you know, obviously they're searching for biochemistry documents.
You want good biochemistry results. The user is expecting biochemistry results.

So before that can happen, we need to, you know, cluster certain documents and call them biochemistry,
certain other documents, you want to call them astronomy and so on.
Then the query that the user provides tell me about black holes, and that's an astronomy quake.
And hopefully that query called tell me about black Holes will actually then take you to the black holes cluster.
That's again, exactly pretty. Believe it or not, they're all related to an idea called embeddings.
Say embedding is simply this some multidimensional vector, simply a python list with like any number of values,
maybe 100 values, 10,000 values, 5000 values, a simple, one dimensional array, floating point.
Come on, floating point convert imprint becomes that one dimensional array.
That list that python list is a vector. Because one of that, the first element is X-axis, second is y axis, third is the axis.
If you think of it like that, the whole list then becomes like a big arrow and the arrows tip as that point.
So multidimensional point. So that is what is generated by this idea called an embedding.
You need an embedding, you know, like a layer embedding algorithm,
embedding a neural network even to take raw data like a piece of music or a sentence or a whole PDF file in audio in one word maybe,
and then put that somewhere, meaning embed that in their multidimensional space.
Then after that happens, hopefully because scientific terms that talk about astronomy is not the same as scientific terms, to talk about chemistry, you know, or about love, they're all different.
So the idea would be then these plans, these documents, these URLs, this content will naturally start to cluster with each other.
In other words, all the things that talk about law would go to some part of this multidimensional space.
Now the da da da da da da. All close to each other very far with a nice gap between them.
All of the documents talk about zoology, but animals would go to some other place in the same multidimensional space.
But now the dots are all clustered.
But these two clusters have a big gap between them because zoology terms usually don't have anything in common with astronomy terms.
Okay. That that's why.
So it's all about how human usage, you know, in words, in scientific words, there are very specific ways that we use it in words about chemistry.
We use a very different bunch of words about data structures.
You know, they're not the same. And so as long as that is true, there will be clusters in the world.
And then clustering algorithms can very easily find what those clusters are.
So clustering is all about finding boundaries between clusters.
You've got one cluster here, one cluster here visually, and look at it in 2D and go out look.
That's a big gap between them. That's why this work. So then clusters can be named.
So you will name one cluster astronomy cluster and then one cluster. You're not going to order one that that will be followed by classification.
So you have to cluster before you classify.
So classify means you already have clusters and somebody is trying to add a new document to this whole bunch of clusters.
So what should I call the new document? Chemistry Document. Physics document.
Gonzo would be in the multidimensional space. Find out where the new document lands.
And then what are the clusters? Call it that. So we call that classification.
Great call. So classification is nice. And once classification is done, it can also be used to answer users queries.
So in the user queries for something like I want back commissioner documents, you can do something very interesting.
You can give the user a bunch of documents that would be biochemistry and also non biochemistry, and then ask the user you told me which is relevant.
You tell me what is the biochemistry and the user would obviously know right there said this biochemistry, this is not biochemistry.
Then what you can do with that feedback, That is where the whole feedback comes in from the user.
It's called relevance feedback. You can then modify the user's query.
The query is a point that move the data somewhere else and that somewhere else is even more amazing than the previous dart location.
Previously, the dart was ambiguous. It gave it.

If you did not leave that, if you did not modify the previous location of the query, the user might have gotten biochemistry documents.
Also irrelevant. Non-bank or major documents never had your search engine.
So instead, now you can actually move the query like a vector, go that way and then start answering.
So that vector is now driven towards more biochemistry results, if that's what doing so called to query modification algorithm that is called Hill.
I want to tell you about Rocky algorithm. It's a query modification. And after the query is modified, user will get amazing results and they happy.
But the claim modification is coming from user feedback.
Chad Jeopardy does not say horrible things to you because.
The standard language model after training would say the most horrible, vile, evil, gross things that make you throw up.
But then what happened was humans were then given ten search results for each grade that you ask, you ask something very graphic.
Tell me how to cut a child up in pieces. Okay. And I'll tell you. And then they say, tell me like ten different ways and how I could do that.
And you show all the ten ways to a human, sadly, and ask them in terms of how discussing it is ranked them from 1 to 10.
That is called human feedback, reinforcement learning for all.
And sadly, people who are paid one $2 an hour in foreign countries like Kenya for this to happen, and then you punish them.
So then you told them, you know, these ten that rank them.
Even the 10th rank don't investigate this because horrible. So that is from a reinforcement learning account that called guardrails.
So that is how l'oms gradually are trained not to say like very horrible things.
How do I make a bomb?
How do I kill the, you know, the precedent, All that if you can ask, but I won't answer you because it's been like taught, so to speak, not to answer.
So that is a form of reinforcement, etc. That's a form of human feedback.
Her IQ is also human feedback. So you'll be Iraqi.
It goes back to 1970s. Okay. So what you consider so cutting edge, 20, 23, 2022, they go back at least 50 years.
Likewise, all of us in this whole lecture business so many times done frequency, inverse document frequency.
They're all just vectors in some abstract space. So machine language embedding is simply nothing but vectors and multidimensional space.
I can hum a piece of music theater to do installations Beethoven just from a piece of music,
because my music can be embedded into a multi-dimensional point, or they're just humdrum.
And writers Beethoven's actual symphony sitting there.
It'll find that in the longest Beethoven like Shazam, you know, for example that we can identify tunes that you home, they're all the same idea.
It's a very powerful idea. The idea is all about embedding. How do you take whatever music, you know, radio humming tunes, architectural diagrams,
chemical molecules, and just simply embed them in some abstract space?
Then you can do what is called similarity search, which is what the whole cosine distance, all that is.
Okay. Okay, so cluster then you classify. Let me show you.
A string or hear that quote.
So, you know, our clustering is about document clustering, you know, because it's classic search.
They go on the web and you said something or you go to a library and research for some books.
But the same idea can be you can remove the word document and then say music, clustering videos,
clustering, you know, animated movies, clustering photographs, clustering, art, clustering.
So all cluster cluster clusters, same idea, right? So then it's all about representation, which is, by the way, just simply a multidimensional point.
So I can even tell you what the even means and success criteria.
You know, the cardinal like what is a good clustering.
Well, and then when you class and you've seen this before in data mining by the way, so in machine learning you have a

classic clustering algorithm.
So for example, camarines clustering, that's one of the things I'm going to tell you.
You know, you can tell the whole bunch of documents, cluster yourselves into four clusters,
you know, our five cluster, six classes, that is K that's one algorithm.
A second algorithm is something called a hierarchical clustering algorithm that is not K means.
What it is is you you initially start with one big bubble.
You think the entire thing is clustered into one giant cluster, but that is not really useful.
You know, let me say, let me divided into two clusters, you know, and then let me divide each of those into two more clusters.
That is called a hierarchical partitioning. And then you will get the same K means cluster.
Say K was six. Okay. And I can I can directly make me six clusters or one giant cluster and then divided into two and then divide one of them into,
you know, one, into one of them into one. And it's easily possible to get six like that.
Okay, so two different ways of clustering you can definitely look at.
All right, so what does you in a cluster? Okay, this is a lot like software development.
When you write a class, you know there's a class and a class B, So why do you think there are two different classes?
Because within a class, things that are very related, they're all together.
And there's a difference between one class and another class, you know, So that way then the two classes can be loosely coupled.
Likewise, in a cluster within one cluster, all the data, all of the documents that are physics, for example,
they're very close to each other and any one of them is pretty far away compared
to all the documents in a different cluster that's big intra cluster distances.
Okay, So that is what it is. So within one cluster, any one cluster, all the documents should have similar words.
Likewise, when you take two different clusters, there should be like massive differences between them.
So that is the assumption. Okay, sometimes it's not true.
Sometimes when you have a photograph you say, What should I classify this photograph I uploaded to stuff?
You know, what should I call it? I can call it travel. I can also call it artful photography.
In that sense, it is not unique, one or the other. Yeah.
So then in those situations, these kinds of questions actually fail, meaning suddenly your document is two different clusters.
Here we assume that everything is only in one cluster. It cannot be in more than one cluster.
Okay, I can even draw a little quick picture so we can have something to look at.
Suppose these are all some documents about something.
For example, algorithms, algorithm, time and complexity.
These are algorithms about web servers.
Okay? Web servers, the night areas,
all the things that you say about Web servers in general might be very different from complexity analysis using the big notation.
So then that way this a bunch of this, a cluster of documents and recloser documents,
and it all of them is going to have similar things that you say for example HTP versus keeps TCP IP,
you know, reverse proxy DNS cache, you know, browsers, handshake request response on and on and on.
But those would be so different from complexity analysis, you know, and log in cost, amortization,
you know, whatever, you know higher skip lists you know to do like multi level linking.
So those have nothing to do with this terminology. Right. So that's the idea of clustering.
So if that is true, then things can be clustered like even even the one here that is pretty close to here
know between two classes even they would be like so far apart in terms of content.
So that is how clusters work. Great. So like this I told you then.
So clusters are nothing but unsupervised learning.
In AML, there are two Broadway, two tier two learn learning algorithms supervised versus unsupervised with unsupervised.
These things like semi unsupervised, you know, and so on. Right. But that is all still supervised but radically different is unsupervised.
Learning and charged speed by the IGP is actually unsupervised learning.

For the longest time we actually ignored unsupervised learning was like a simple step because,
you know, because supervised learning was the most magical thing, like neural networks.
But clustering has always been a prime example of unsupervised learning.
Unsupervised simply means this when we have these documents, you know, in 2D, for example, suppose I give you a whole bunch of X come my way.
So each any one of these points anywhere are just simply x i.
Y meaning zero i. You know, when we were next to I do all the after study where study just simply a collection of two dimensional points.
You can just on random numbers actually in a square and just say, you know, randomly generate points like thousand points.
So it's going to generate randomly thousand points, Right? That is all the input is.
It's all input is pretty much how many of the points you want to point to.
And then you can tell the system, give me three clusters.
Give me ten clusters that you have to specify. Now, actually, you enter this way.
But for now, we'll just say you have to specify that in this case you can say, give me three clusters.
And what might possibly happen is there might be a boundary here and then maybe there might be a boundary here, you know.
So then I made three clusters sort of them. But that's all input. You don't say anything more.
That is why it's called unsupervised. Then I think what unsupervised means and supervised means this again, you have data.
I'm going to draw a table. You're still have data where each point is data.
Data, data. I'm going to make role for each point. I make a role for each point.
Row, row. But now suddenly, I'm going to say something called the Y axis.
In other words, maybe I have like x0x, y, y here.
So as usual, I take any one point and I record the x coordinated y coordinate.
But now I'm going to say something additional. I'm actually going to say things like bottom left, bottom, right top.
So if this point was that point, the first point was this point, I'm going to say cop, I'm going to call this a class.
Okay, Class. Usually we call it I call it the y axis, but I'm going to call it class four now because that's what you want the classification to be.
So all these points, right? When I make row for each one of them, I'm going to say top, top topped up any one of these points here.
I'm going to enter the x y value and I'm going to say bottom left.
Any one of these points in here, I'm going to enter right here and say, bottom right, these don't have to be grouped by the way.
You can scramble them. How are you want? But that last thing that I told it, it's called supervision.
So now I want the machine to know that any point that is in this area, I want it to call it bottom, right?
That's the label. I give it to label class as a label,
so I label it and the labeling is what makes it supervised learning because I'm supervising your learning the label.
And then basically what the neural network would actually do is, believe it or not, learn this boundary.
It's pretty amazing just from this table, even though it cannot draw visually.
Look at it right? If you call this bottom right, if you call this top, even if you call this top, you call this bottom left, you call this topic.
All this bottom right is going to know where the boundary between is. Yeah. Yeah, correct.
And even before that. Even before you. Right. But even before you could do that, the clusters themselves don't have to be supervised.
Yeah, crack clusters are not supervised. And the new observation that comes in is simply assigned to a cluster.
Yeah. No supervision anywhere. Whereas here is the opposite. You have to label all the existing data.
New data will also be labeled. And that's called classification. Exactly. Correct.
Good. Yeah. So, you know, that's pretty cool, right? This can be the most complicated.
It's called decision boundary. Okay. It doesn't matter how complicated it is, as long as you label it properly, the machine will figure it out.
In deep learning, the more layers of neurons that most complicated boundary can be learned automatically.
Just do back propagation over and over and over to error minimization. It'll work.

Okay. Okay. It's pretty magical. All right. So coming back, though, that label is not provided in standard clustering algorithms, okay?
Just simply say, give me three clusters. That is why we call it unsupervised machine learning.
You get it? Okay.
Likewise, in Chargeability, the labeling would mean I take one sentence like the dogs barking, and then I would have things like the duck colon noun.
In other words, the word those a noun barking a dog so that there is a definite participle and then dog is noun, barking is a verb.
So I will tag every part of every sentence into nouns, adjectives, adverbs, you know, pronouns.
But we don't do that. The English is like, there's so much to learn.
So we simply give that the all of English to charge it and say, Learn it says unsupervised.
Okay, so jpt and clustering unsupervised, and that is supervised machine learning.
In contrast, now we know so clustering, so called recommendation engine.
So that's also clustering. You know, so all the people, you know, that are all the songs that are, for example,
listen together in Spotify, all the Netflix movies that are played together, they can be clustered.
And then if you go on in the cluster, watch only a few movies.
That story that the rest of the movies that the rest of the movies for you because I usually watched in a cluster why just ask the people request things are also bought together in a cluster and cluster them and it says you only buy one of them,
Amazon, Which are you people? That part is also but are people that viewed this also viewed on StackOverflow.
Your ask a question people are asked this question also asked over and over and over also also also there is also clustering.
So clustering has so many users. Okay, it's Sunni. Okay, then let me tell you more.
Yeah. Related searches like I just know started telling you search related.
You know, you type something up here and then they will expand the search search is related to cars.
You ask about cars, it might tell you more like used cars, cars for sale.
You know, for example, a cars to full movie and also enjoying a Pixar's cars in all those workers.
So then those kinds of related things that do this also to for example you know what is what is async in JavaScript or something.
So that people also ask that because all these words.
Right, are all clustered right next to my query, because going to find all that said last time,
inquiring verbatim meaning is trying to find those words in in my query and then that is what resulted some kind of snippet to me.
But then that query is very near all of these embedded planes as well.
So it's extremely useful to us because you might not know that that's a different way to formulate your query, but if you click on all of them,
you'll get slightly different links, which means if you look at all of them,
you will get a much better answer than just reading only the links from your query.
Meaning these probably are the links from my query. Right. But then I should also maybe go and look at all this.
For example, if I ask what is async right? The next question is why do I care?
It's pretty neat showing that to me, right? Is expanding a knowledge, you know, why would it not be useful?
Okay, that is all clustering. Everything is clustering. All right, so Peter column, I'm going to skip some of these older ones, you know, So yep.
Was actually a very smart idea because they understood that clustering is pretty amazing.
This company called Innovation animated. Any search automatically came out with clustered results.
They literally clustered the results. You know, that's so cool.
And in a library, all the books that you go and search for in the stacks that are already clustered.
So you know that that's the whole idea called epistemology in what's called epistemology.
So epistemology is basically the study of knowledge of knowledge.
So knowledge in the world has always been cluster slash classified forum.
So when you get a Ph.D. doctor of philosophy, so philosophy is supposed to be ultimate knowledge and knowledge about the world.
But these days, so our best days are pretty steep.

That's when somebody says or appears to usually you don't ask them in what at some level is supposed to be irrelevant. Okay, because all knowledge anyway. But in reality, because subjects are so different, somebody would say, I got a pierced in biology.
That means they studied something very special. What biology?
If you take this abstract idea called philosophy and a group them philosophy can be about God,
can be about religion can be, but society can be about deepwater.
You know, aquatics can be about military tanks. It gets very, very specialized, more and more like a big hierarchy of knowledge.
Okay. So in that sense, clustering is very useful to know exactly.
You know, your query matches what other way. That's why in a library, if you walk around, all the math books are in one place,
but even written books, all the topology books, all the trigonometry books.
One time I told you, Dewey Decimal classification.
So John Dewey was a philosopher, so he made this incredible way to classify practically any knowledge in the world.
I mean, look at that. This is like super cool.
So he basically said anything at all that you want to talk about, and the entire world fits within these big numbers.
And within those numbers there are even smaller numbers. For example, for two might be Chinese and then for 2.1 might be Mandarin.
So you can have a decimal system and decimals can be in infinitely. And I did write more and more decimals, so the classification is endless.
The top level category is fixed forever. We still use this, by the way.
So any book at all in the world, if you look at the book, the first two pages, it'll tell you what the Dewey Decimal System is.
It's so cool.
Okay, so all this is again, related to what I'm saying to you, which is, you know, this notion of clustering knowledge, it's always been valuable.
I know our search engines get do it. I tell you, Yahoo! Failed.
I mean, I knew that all of the people that worked at early Yahoo! In religious conversion.
So Yahoo's idea was so different from Google's idea, but Yahoo's idea was a clustering idea,
manual classification, you know, So if you know what a web page is,
suppose the Web pages, PepsiCo, then they'll put that under junk food beverages and how they'll classify property dot com and the junk food beverages.
If there was a site called Nasser that you know JPL dot gov and I said I'd go
that to classify that under astronomers census so that is in all classification
which means clustering here human derived clustering but that failed only because
they could not keep up with all the billions of pages that were published. Not that the clusters were incorrect.
The clusters simply followed the Dewey Decimal System, but there were so many pages that nobody,
no human could possibly know manually placed them every single day just to give up.
So search was a great idea. So I simply said, Well, look in the document itself and find out know what it is.
And basically to try to find it on the user doesn't have to care about like what it is search.
In this case, if you're searching Yahoo!
So you're looking for a biology book, you would click on biology, then you to click on botany, then you would click on palm trees,
for example, you know, But now in Google you can just type palm trees and all the documents magically come to you.
In that sense, classification basically failed because it is an unsalable idea, it cannot do it and cannot keep on doing.
Let's get back to clustering. A good cluster again is intra class within one class, meaning within one cluster.
All the documents are similar and then they're all very this cluster is very dissimilar compared to documents in any other cluster,
which is what this whole gap means.
So, you know, if you bring them right up to each other, right, and suddenly you start to see that these clusters are not that well defined after all.
Like this one almost could have been this one, you know, but somebody arbitrarily decided it's got to be non.
So thankfully in the world it doesn't happen that way. You know, that's the point.

Intra class similarity in terms between similarities lo and an intro class.
Similarities? I already told you this. Okay, so then that's pretty much it.
And then. Yeah. So how do you represent the document is usually tfidf that will then determine like all of this,
what goes and what cluster and how do you measure similarity, how even calculate distances.
And I tell you the short answer, Euclidean distance calculation squared x0 minus x one squared.
I'd say this is like 00x11. I simply ask you the simplest question in the world what is the distance between them?
X0 zero excellent y and girl at the current time, right?
Because the delta is like 0x1 -0, the delta is y one minus y zero.
So this one is clearly square root X0 squared x0 minus x one squared plus y zero minus y one square.
I mean, it's the most obvious thing by the way, when you square it, you've got to find the square root.
It's called a tool Norm. It's called L2. Norm and L2 norm because there's two there.
You can have an L one normal going to all of that. You can go and have an L three norm By the way, you can do cubes, do a cube.
Okay. Then suddenly it gets into a great curvy space, you know.
But then for a standard Euclidean flat space two is actually good enough.
Okay. Okay, then that is what is meant by similarly measure the distance measure.
So the easiest is to understand visually when it's Euclidean,
except this is totally right where this one where another word, you can have 50,000 dimensions.
Look, I still find this, but except you will have 50,000 of these still all square, and we still take one giant square.
Same idea. Okay, then you will say something like, you know, a square.
Ixi minus y a in a square, but I itself can now go from one to any dimensions.
You know, basically sum this quote. So I'm going to show you like all of that.
All right. So, yes.
The clustering should be, the algorithm should be when you add more and more object to the cluster and suddenly the cluster cannot like,
you know, become like unstable, meaning something at some point here suddenly should not be misclassified.
As you know, the document you need like a good clustering algorithm and k k means clustering satisfy that property pretty easily.
Okay, so it's going to be stable. UNSCR add more and more documents to any cluster is still going to be properly, you know, cluster.
Likewise, if your chain, if you make small errors and where these document locations lie, then it'll change the cluster just a little bit.
Meaning, for example, this one,
I move this slightly next and where it's going to still be in this cluster that's also considered stable that such an algorithm is stable call.
And also it doesn't matter in what order you consider all these points to cluster them.
It's completely competitive in non noncompetitive non anything order doesn't matter because
in the end you're going to get the symptom element cut to the chase for your is very simple.
When you cluster something, what you're looking for is one magical centroid,
a cluster centroid, which is an excuse my point, the central center of gravity.
Likewise under the centroid, another center to centroid is what we're asking the algorithm to come up with.
And that's an iterative algorithm. There's a pretty cool, easy algorithm, which I'm going to show you and talk about it.
So when you want re centroid,
so many things in machine learning from a data mine can start with completely random numbers and then we can loop just to a like a while loop.
Gradually it'll converges back propagation, generic exact same thing.
Make all your neuron weights be python random numbers between minus one and one completely horrible.
But over time, because with our back propagation,
those initial random numbers for the weights will slowly get to be the actual weights and then magic happens.
Likewise, these three centroid can be entirely random.
I could just close my eyes and say, you know, I'm going to be one centroid.
Second centroid doesn't, right? I've no idea what happened. Right.
Those are centroid. I'll just call it the one centroid. Second centroid, that centroid, because I know I want three centroid.

But what'll happen is by magic, we can make that centroid move to that real centroid.
We can make the centroid move to the centroid, the real one. We can make that centroid more to that centroid through many iterations.
And then you have what you want. It's guaranteed to converge.
Guaranteed. It doesn't matter what the initial starting points are, they'll always go there.
Okay. And I'm going to show you a nice visualization with the spacebar and go like little smoke trail.
We you know, it always works. So I'm going to show you.
Good. Yes. You know, again, in the classification, you already have.
You know, so regardless of the fact that clusters don't have a name after it is clustered, you're going to call it something.
In other words, I know that in my big bunch of documents, you know, there are definitely physics books, chemistry books, math books.
Find me three clusters, and then I'll call those classes. Physics, chemistry.
And that is how when the user searches for something, I know, like, what to serve them.
Okay. In that sense, yes. So we do have, you know, a bunch of predefined classes.
So clustering again is when I give you, like all of them together, you need to find those boundaries.
Okay? And that's ultimately what it is called. And like I said, the unsupervised learning and all of this.
So clustering versus classification against clustering, unsupervised classification supervised,
because after you cluster, you can assign labels to the cluster. At that point, the result then becomes like a classification task.
So one leads to the other cluster first and you see results clustering.
Great. One more. Yeah.
So clustering this what clusters look like in just two dimensions?
Abstractly. Suppose you want like, five clusters or something.
Then you can imagine, you know, in some two dimensional space, the clusters will look like this.
Again, the whole idea is there's got to be some gap between, like all of these individual clusters.
Otherwise, the whole idea becomes meaningless. Okay. There's gaps.
The shapes of them are actually very fascinating. Um, you know, ideally, those shapes are like some kind of weird spheres.
Okay. If you have lots of gaps between them, not a gap between them, then each one can be like a spherical soap bubble, you know?
So each cluster is also bubble. But when the circle will start to grow towards each other, then they'll actually, like, run into each other.
And so then this these curves, right, would become a straight line. That's going to become a straight line.
There's a big my state line. Straight line. Straight line. They'll actually meet at a vertex here.
It actually start to get convex polygons. Okay. That is called a very diagram.
And I tell you about that as well.
But then ideally, you know, if that is not the case of the cluster that far apart, they would not run into each other.
Then you can imagine that was breaking down. Okay. In other words, the cluster has a center somewhere,
and all the documents that lie within a certain radius are going to call it like that particular cluster cluster one.
All the documents that lie within the second radius, you're going to call it cluster two and so on.
Okay, so it's all about centroid. Anyone ready? Okay. All right.
So again, even though we said it's not supervised, after you form the cluster, you must certainly want to call them this.
So think of this as after clustering. In other words, initially you cluster them and then called a certain call, cluster certain names.
You know, you know what names they are subclasses of name. Yeah.
So then those boundaries. I told you those boundaries, if the crosses are pretty far apart,
would not have these ellipsoid boundaries, you know, because this radius is a big radius.
And then you think that the radius covers like you in this ellipse. Meaning I'll make the ellipse into a circle.
Okay, That is what you give that far apart. But then in reality, you know, they're not that far apart.
And LPN, I. Oh, my God. I mean, this has areas actually, you know,
look one of the one of the subbranches are I would be an LP like you know sentence classification
the sentiment analysis like all of that right So there's not that much of a boundary between them.
So there's an LP data that would come all the way up to this boundary.

Meanwhile you have the circle. I would just include things like reinforcement learning expert systems.
Then that would be on this purple, you know, that, that they have a linear border between them likewise and LP and graphics.
I want to visualize, you know, words, I want to visualize sentence structure or something.
Then all the gra, in other words, this is 3D rendering, you know, maybe ray tracing, all that'll be over here.
But graphics related to like you in the new generator I characters that actually automatically took the non playing characters
in your game there would be graphics and then they're going to come towards each other like quest Graphics and architecture.
Wow that's my whole world. And Nvidia GPUs. So JP would be a computer chip design and also 3D rendering.
That line is going to be the boundary of computer architecture and theory paging algorithms, caching, you know, branch prediction, all of those.
Okay. By planning, that's like theory as well as architecture, like wisteria and wow, the roots of deep learning.
So what makes deep learning successful?
You know hardest it and learn arbitrary boundaries because a thousand page book by the vocabulary at some point the link to it.
Yeah operating theory is crazy for the longest time until quite recently when no idea why the [INAUDIBLE] the planning work.
At some abstract level we knew. But then somebody actually made like a very amazing document.
Here's a summary. It's all about the shape of the actual deep learning architecture.
The shape is the the biggest determiner of why it succeeds.
Okay. Okay. So anyway, so they all become lines at some point.
So let's move on. Yeah. So classification requires clusters and boundaries, Right?
All right. So this again, you know, by the way, these slides, if you if you notice what happened is we talk we start with clustering, right?
But then suddenly we have gone on to actually talk about classification as well.
So don't be confused with what we're saying. All this is going to be true after a cluster will come back to clustering.
And I'll tell you about came innovation. All right.
So in classification, obviously after you cluster, you label the cluster, then you're going to call it like a training.
Okay. Even though we didn't do it this way, we did it a little bit differently.
We clustered and then we call the cluster something. In other words, we did the labeling, whereas in supervised machine learning,
the labeling is already part of the input data and then it learns where the boundary is.
That is a difference. Okay, so we label them, we still have to label them, so we label them afterwards.
The new query or new document is coming in and all you have to do is find out what circle the new query are.
The new document went If it's a query, then return all the other documents as a result of the query.
If it's a new document, call it whatever. Say this all fix documents.
A new PDF came in, call it physics, that's all. So it's simple difference between what is a query and what is a press classification quo.
Another word squeeze or answer. Using classification, you go in circles, but that's what it is.
So again, this to to the ultimately, yes. So then how how or why are the documents why are these dots close to each other?
Because in the term frequency inverse document frequency space in that mathematical token space tokens that relate to different similar documents,
meaning two different astronomy books would be very close to each other in that space.
So then it all comes down to exactly this.
That bag of words again simply means I take out all the stopwatch like and or not, and I have done all of that and only focus on the actual nouns,
mostly in the sentence, and then find out where that whole sentence ends up in this embedded space and summary sentence.
Also very similar about changing car batteries in your car. All the all the car battery changing manuals would basically say the exact same thing.
They all say, make sure the car is turned off. Make sure that you connect, you know, the positive to that kind of ground.

Okay. You're going to get a shock. They'll tell you all the different same thing over and over.
So then all of those would become close to each other in some kind of extra space and that's it.
So then once you cluster, you can classify it.
So classification is simply the algorithm that says,
I'll take my new query vector and then do a cosine similarity between some kind of nearest neighbors
by doing this also where the whole idea of vector indexing search comes in like damn important,
supposing all these vectors already all these dots are already clustered, you know, say there's a cluster here.
Okay, Cluster cluster cluster. But now say a query goes here, my credit goes here.
And I said to you, I should serve the user nearest to the query documents, the basis,
serve the user by relevance, highly relevant, little less relevant, relevant little restaurant.
Definitely don't serve them.
Okay, so I'm sorting by distance or by angle, even documents that are pretty close to this document and also documents that are pretty far.
So again, how are you going to do that? There's going to be billions of these points, right? You cannot in real time search through all of them.
It cannot do distance calculations. So we do something pretty incredible.
We index it, we partition all these dots and then to hierarchical partition,
so much like region three in that space, indexing maps load very fast because the map knows where your car is.
So if you say Indian restaurant near me, tie for near me,
it'll only load Thai for the restaurant within like a timeless color tile because you already indexed the space.
Likewise, you already index all these dots and there are so many indexing algorithms face this.
One of them FSS face vector indexing algorithm, also hierarchical, navigable small space.
I meant to say all this in like two weeks, but then I'll just tell you right now, size versus hierarchical,
navigable small worlds on these are all just simply indexing libraries, that's all.
And you can simply use them. So after indexed them, this point can be really restricted.
Only this little section, it'll never look over there at all. So that's how it's radically fast.
Okay. Okay. So these are pretty neat. Yeah. Okay.
It's all about this. Like this. You know, you can even read about, like, how they're all different.
They perform, but all that doesn't matter to us right now.
Just know that when a new query comes in, which is a star, what are you going to, you know, serve the user?
This similarly, it's about. Okay. Seniors never say over and over.
So that's really all it is. So how are you going to search for nearest neighbor like that?
Again, query image in art in this case, see how sunset pictures are in the data set and then you draw like a pretty cool looking sunset.
It'll actually pull out sunset photographs. So it's a form of image based search.
I can draw something that I want. Okay. I can go on and on. Nvidia has an interface where you can draw very crude looking mountains in a waterfall.
It'll turn that into a gorgeous piece of art. It's called Gorgon Geology and Gorgon.
You can look at Gorgon. Okay, so same idea. Okay.
It just ultimately decision space is partitioned, although that means, as you know,
this in space again means where the clusters are after your partition,
after you make the clusters and you indexed them in, the query becomes really easy.
So this the whole idea given a new document, so a new PDF comes in, what should I call it, or given a new query?
New document, new query. They're all the same goal ultimately, or some new embedding that the system is not seen before.
That's all. Okay. The question is what is around you or what cluster did you fall in?
The same questions over and over again. All right, So now we can come to clustering.
In other words, how do all these clusters form? If I say give me three clusters or even five clusters.
If I say give me four clusters, you might do something pretty weird. Like, look at this, right?
I told you, there are some points in here, right? But there's some gap here.
See that? There's some gap here. So it might even make a cluster out of this.
It'll give you how many clusters you want. Okay. That is why you should not ask for way too many clusters.
It becomes meaningless. Really. It's like know random division or you should not have too few clusters.

Give me only one or two clusters. Just an ideal number of clusters, sometimes trial and error.
Okay. All right. So how do you represent the document vector space?
X-rays, the axis term frequency, inverse document frequency.
How do you compare similarity, distance, cosine similarity.
All that that means is and I can draw that dotted here if your query point went here, grapevine one here,
and then there are some other documents that are already, you know, in the system, so to speak.
Then look at this document that is also like a vector, right?
This angle between them is small, this data, whereas this document has a bigger angle.
This document here is physically close and also a smaller angle.
This document is even smaller angle. So if you look at the angle, the angle similarity, this is a pretty big triangle compared to my star.
This one is also a bigger angle compared to my star. You get the idea.
Okay, so forget the distances. Just go with angle. You want to find out the closest angle related documents because cos of zero cosine of zero is one.
It means if the angle is brick small, almost like tending towards zero the cosine.
That's why it's called quotient. Similarity would be a very high number if the angle is pretty large.
If you are Star Wars over here and there are some document literally like you know in here for example,
you know like over here some documents, that angle is almost 90 degrees, right.
Cos of 90 zero cosine of 90 zero to call L cosine curve cos zero is one cost 90 zero and then costs two 7360.
Just keep going back. Quote So cos zero is one.
That's exactly what it about. And of course, 99 degrees pi over two is zero.
So there's a part of me care about. Okay, that's well, so I'm going to show you that right.
And similarly, how many clusters, Right? Exactly. So, you know, this is usually known as a fixed API or sometimes you can even do it automatically.
There's a cool algorithm called expectation maximization.
And so data mining algorithm can m it can actually come up with the serial number of clusters.
You simply say cluster and I'll give you a what, the proper number of clusters, or you can do the following, can do an experiment.
You can actually plot how many clusters you want and basically, in other words, ask for a whole variable number of clusters.
Give me one cluster, two clusters, three clusters, four, five, six.
And then here you can compute some kind of an error after the cluster has occurred to see if my clustering one properly or not,
I can quantify my clustering with some kind of error that I would do something pretty interesting.
That error would initially go like this, for example, when I only have one cluster for everything, right?
It's a pretty big error.
When you have two clusters, the error would fall when our three cluster would fall here maybe, and then there would do a bend.
When asking for clusters as a band five cluster when that band is the ideal number of clusters.
So we should be asking for three clusters. That's called elbow room.
Okay, you can do an elbow rule and you would know how many clusters you want.
That is the answer to all of what is the ideal number, you know, So if you don't notice,
you can actually do an elbow roll to find out or we can let expectation maximization do it automatically for you.
Okay. So I don't know, cluster count, cluster count, data mining, elbow method.
Well. That there's an elbow point, you see.
Okay. Elbow point to find the optimal number of places. I mean, they can go read like so much in on your own.
So I'll just simply point things out to you. There it is.
Okay. Oh, that's where that came from. Okay. Maybe something.
Maybe one more. Something simpler. I mean, you'll definitely see it.
Okay, Because it falls like that and there's definitely a change in the slope there. And so that is the ideal number of places over and over.
Same idea. Okay. All right. So you can go right about that. Cool.
That's how you know. And then like I said, just now, don't make just one clustered.

Everything is pointless. Also, don't make those in process. This. I'm going to switch by Goldilocks.
Okay. So that's where the whole ambiguity comes in. Hard clustering is when each document falls exactly in one topic.
I know this a math volcano that's in biology book, but what about mathematical biology?
Population dynamics. Don't talk to me if you do that.
That's called soft clustering. So in soft clustering, a document can have more than one cluster.
It is a difference between tagging and categorization. That is why tags that became so gosh darn popular.
Okay, hashtags. Because before hashtags, when you upload a photograph, you've got to pick from a dropdown.
Tell me, is it a photograph about cooking? Is it about travel?
Is about photography? You got to pick one, look at a job and only apply.
You've got to pick one of them. But what if you say can apply for more than one job?
Sorry. Two. Very good. Pick one that is work. Classification is hard classification, but tags you can put in how many or the tags you want.
And so then when people search on any of those tags, then your stuff would come up.
That is a multi label classification. This is how clustering quote, for example, you know,
if i search for L.A that the local news should come up and you maybe national news like somebody in some
other state searching for ella should still be able to read the ten freeway got lost or the we can,
by the way, a piece of ten freeways bottom car on fire. I mean, it's crazy.
It's almost as bad as a 1994 earthquake. Imagine closing kennel.
Okay. Oh, my God. Okay. You are near USC. You can feel that heavy traffic because that is insane.
All right. It takes 11.5 X times for me to go home.
So that should be like national news, right? Actually, that's like white sneakers, right?
So where you want to sell sneakers, then? Another manufacturer, what should they do?
You do it under sports listed in the sports apparel also shoes because sneakers are issues practically.
That's why that matters. Okay. Variation in our store listing or listing, you've got to put in what shoes tags for that.
Okay. Now we come to the actual part is so simple cosine similarity over and over and over.
We send this before the dot product, a vector called a and a dot product, electrical B, So eight are B is just simply computer graphics.
We do this all the time. My nose has like an arrow going out. It's called a surface.
Normal light has narrow coming down. It's a light direction.
So you can predict product. That's what it does. Illumination. If my nose directly looks at the light, my north tip is bright.
The side of my cheek doesn't is finding this way. And so that part of the slow, lower and becoming darker.
Okay, let's create a 3D shading. So recall the same idea.
So just one time I'm going to write this for fun. If you have a vector call, a similar vector here.
Colby The idea is what is a-to-b? So you want to minimize so as your query and B could be some other random vector, meaning other documents.
So you want to return the document for which A.B. says as large as possible, error should be as low as possible.
And that is the internal domain by the angle, because, you know,
A.B. is just simply length of a how long This is times length of be, but they can be somewhere else.
The same B can be here as well. Say, B, the length doesn't matter 18 times cosine of the angle between them.
That really is the biggest determiner because then if I spin, be in a circle,
if B gets pretty far away, in fact the biggest 90 degrees that that product will go to zero.
So the cost cost data is what is determining everything further.
You know, we can always normalize. We can always take a and divide by the length of air that's actually was done in the slide.
So then that is a in fact, that's actually what this length of eight divided by.
Actually in unity. So what you can do is turn ten.
That ain't where like a hat, you know, a divided by length of a and would automatically be a unit vector called a length one.
Another unit vector called B, also length one. Then it's going to be simply one times one times cost data.
So the length just goes away. Length is irrelevant. So all about the cost data.
All right. So then that's what all of that is showing up here. I can go up here, call, you know.
Okay. So what about the actual dot product itself? That's very simple.
Right? In analytical geometry format. Say it was called EC zero zero said B vector was called x one, Y one.

Then that product as you know it on B is just simply x zero times x one plus Y zero times y.
One is simply multiply that X's and multiply twice and just add them.
It becomes a simple numerical value becomes a scalar, right? Becomes scalar crossbred.
It is not a scalar. Cross product is going to be another vector. Okay, but this is terror.
Okay. That is all this is. So when we say air dot by it is simply x zero in our times y zero because this is like 012.
These dimensions say zero sum multiplied x zero times x one plus y zero times.
So I want in our case, but in 50,000 dimensions you can do this particular times.
I still become one number. Okay, a simple number and that number can be normalized.
Okay, then that's actually what is done here. So the simplest one, this is a unit vector.
Is this a B, and length is one. Therefore, it's all about the cross product.
Sorry, it's all about the multiplication quote. So then we use question similarity, but it can also use actual distances.
So maybe a different way to find out what should be served to the user is not use the angle at all, but actually use distance.
So again, my query was this little star. Okay, I'm going to say from the star, how far is this literally, what is the length of the ruler?
How far is this look? How close is this? How far is that done?
How far? How far? How far?
Then I can sort them and say that one that was so close is my first relevant, highly ranked result and then my second rank, third ranked by distance.
That is Euclidean measure. And even there there's an alternative called our taxicab L1 measure.
I'll say that now. All right. So again, you know, set and vectors means and dimensions or can imagine the sentiments in the 50,000 dimensions.
Then how where do you find tree clusters? In other words, how are we going to make it visually?
You can see that recklessness, right? I'm going to show you the that came in them really simple.
So in Camarines Sur, we definitely use our Euclidean distance, but you can replace it with other distances.
All that will happen when you change the distance measure. Here's the cluster boundaries will change.
In other words, say initially this was considered to be that cluster.
It might become that the boundary would go over here and this might become like a second cluster.
That's the practical result That'll happen. Okay.
And it doesn't matter what what kind of a distance measure use if there's lots of gap between the boundaries.
But if the gaps are getting smaller,
then the distance measure that they use would affect exactly what kind of class or what kind of a cluster you would end up in.
Hey, great. So move on. Yeah. So again, partition, hierarchical.
In other words, partitioning is that means I'm going to show you an hierarchy.
You simply start with one giant cluster and then refine or go the opposite way.
Tell yourself that every single document, each one of them is a small cluster of size, one little micro clusters.
And then depending on how one micro cluster is close to some other, micro cluster rise is far away grouped down.
And so, you know, now we have a cluster of size two, I have a cluster of size to have a cluster of size to size two, such to size two.
Then these cluster of size twos are closer than a cluster of size to here and a cluster size two over there.
So merge them and make cluster of size smaller. I can go from small to large.
I can go from bottom to top and stop exactly when I reach my how many clusters I want.
I can go from top to bottom as well. Okay. Yeah.
So top to bottom is actually better than what I'm. We're going to get that impression.
That is what this is. So in here, you're going from bottom to top, meaning from one, each document being its own cluster, all the up to the top.
Conversely, you can start with the entire thing as one big giant cluster and subdivide, subdivide, subdivide.
Ideally, you should get the same result in both cases, but in reality you actually will not.
Okay. It's going to be a slight difference between them. Okay. In the meanwhile, I want to show you a partition is actually a lot of fun.
All right, so now Kim means care partitions. Okay, so, Kim, it's cool.
So choose random items.

Huh? Interesting. Okay. Yeah. You know. Huh?
Actually, no. I probably wouldn't even say that. I would say I'm going to say the entertaining differently.
Okay. Please ignore that top line. Instead, actually, what it is is this.
You want to know when all is said and done. The circles are not there.
What do you want? Is those circles, you know, indirectly.
Meaning you want a system to tell you there's some kind of a centroid because you want three present, right?
There is a centroid. And then here's the centroid.
Here's the center. That's actually all you want. We want three means basically.
Okay. All right. Not three medians, not three modes.
That is all what is called summary statistics. Just to remind you, we just want the mean here mean because things like median don't make any sense.
You know mean. It's only because that that is what pulls all of the documents so to speak the mean it's really almost like our
church and then all the documents are like parishioners that live near the church this a different church,
like a competing church across town. And these are the people that live near the church.
So then each church is asking, who are my parishioners? Okay, so I'll be single assigned person.
This person is going to be assigned to their closest church, not to this church that is pretty far away.
Not that church. That's what it's all about. Okay. So mean is what you want.
Of course I came means no. Let's go. Yeah.
So what is happening here is it's all about Euclidean distance, you can see.
Okay, so you want to basically every observation in the cluster, meaning something is a cluster where all the observations,
meaning all the ducks in that cluster, all close to one mean versus some of the mean.
And then that's actually what you want. Okay.
So the mean and we don't know where the mean it's it's a little catch 22 but after everything is done you
want for each for any document anywhere you want a document to be close to one mean versus some of the mean.
So the mean that the document is closest to will be in the cluster.
Other words, for this document right here, that is the closest star distance, not this.
The star is pretty far away. The story's also pretty far right. Therefore, the document actually belongs to that star.
So what then? What would such a story? Then when a new document comes in, the same thing you would do.
Suppose a new document was here, then you would say for the new document,
what are your three distances to the stars, to that centroid, that distance, second distance, third distance.
Clearly assign this to a cluster. Conversely, supposing you document a are out here again, where does the distance to that mean?
Does that mean does that mean obviously send that little cluster?
So I thought that is. Cool. And so then I'm going to show you this whole centered business.
I'm going to animate that very soon. But meanwhile, here's here are different ways in which you can measure similarities.
This is the cosine similarity angle based all the way to the right, simply angle symmetry.
The angle is pretty small between two documents, the highly similar. This is simply Elton John, by the way.
It should be on the top usually. Okay, so it doesn't matter.
So Elton John is down Euclidean distance. You can see square of this means raised to one one half.
If the symbol vanished. Right, it will an arrow. This area is two one half.
In general, this can be p p p dimensions raised to one over 22.
Then it's called l p Normal can be. Can be any dimensions. So in standard space, we used to calculate the distance.
I want to go in front of my attention because now is the time that.
So please tell me this, okay? Please tell me I'm going to die here.
So please tell me that. Okay, So there's a couple of points in here.
Should none of them tell you that? Okay. Just say, you know, just draw something.
I'm going to ask anybody. It's not a trick question is a very obvious question.
Okay. So here, like a pair of points. Here are a pair of points.
I'll ask you which pair of points is close to each other.
Every single person will kid with you until you get to that.

Why asking me that? Okay, fine.

If you ask a so-called A.I., including like a robot racecar, like vision camera, if you ask it which parish which is closer, it'll all be sensory.

Okay? It'll still tell you that it's close. But you know what it had to do?

It had to impose some kind of a coordinate system, meaning it grabs that image, right.

And it's got to coordinate system. You will find this pixel values. It knows exactly where.

It picks up pressure centers. It knows this pixel center right there.

It knows this pixel center and this pixel center.

And I already told it, you know, like I said, this versus this. So it will actually compute the Euclidean distance.

By the way, at one optimization, if you don't have to compute the square root, you can just leave it to it squared.

Okay. It's still the same thing. Okay. So how to do that?

It's going to do x zero minus x one squared plus y0 minus one square here also x zero minus x one squared plus y zero minus y one squared.

Compare them literally to a min or max operation and find the min and then only tell you that's close.

Guess what? We do not do square calculations in our head.

Come on. Okay, so then that is what you should actually wonder.

So why is it so different? That is related to why things like tragic beauty make crap up and why that cannot possibly relate to human emotion.

Like lots and lots of things. Basically, it does not work like the brain at all, regardless of what we call it neural network.

That's complete B.S., by the way, because your neural network is nothing like that neural network anyway.

So think about that. Okay. There's only one way to solve that.

You have to actually make an analog robot and thus not use distance calculations at all.

And nothing like that exists in the world. Okay, so that's a radical change for you.

Okay, so then that is why that's what I was this actually fun.

All it is is I can also find the distance between point and point B, not directly as the crow flies.

Okay. In this case, the crow is flying ratio down here. If the crow flies from point A to point B, literally across, take off.

Right. That's the shortest distance you can ever find. But what if you had a grid?

What if you had a coordinate system? The coordinate system had, you know, all these greater values.

And then I say to you, all the points that I'm going to give you, just this is simplify, okay?

Way I'm going to tell you the difference pretty soon. But for now, imagine I have an integer grid and my points are on the integer.

My point is zero zero, the both integers. Remember, even if they're zero common zero.

That's one comma, one literally. Okay. And this point to something else. This one is one, two, three, four.

And then one, two, three. One, two, three, four, five, six.

So that one is four common six. Okay, that. But now and this I'm going to call x11 will not do Euclidean distance will not find squared.

This one is one. Come on. So will not find the square root of one minus four squared plus one minus x squared can always do that obviously.

Then you're going to get the diagonal right. Won't do that. Instead what we'll do is we'll say what if we can only walk along X and Y?

That's called taxicab geometry. It's called Manhattan Metric because in Manhattan there's all these tall skyscrapers, right?

The streets to go through. So if you have a taxi, you get a cab lift from here and say, take me over here.

That cannot cut across building right down it right on the street. So then you can add what distance it takes.

And that's a cool calculation, right? It's like this. You can say, you know, I'll go like this.

One, two, three, four, five, six, seven, eight, four, one.

I can do it that way. But regardless of what you do, very interestingly, it is going to be this length, which is three plus that length that is five.

It'll always be a sonnet. And so then that is what the L1 metric is.

That L1 metric that is set, there is just simply a different way to think about that is do the Delta X or do the Delta X, which is this plus Delta way. There's a different alternative formulation.

It's called L1 norm because there's no square root, no square nothing.

Yeah, sometimes that is considered more robust, by the way, in machine learning.

Okay. Okay. So then that's where that came from. We use that commonly in one of these.
Second, you can pick any of these measures to do the cluster centroid calculation, like, you know, like where the centroid.
Usually we pick this, this one usually works. I'm going to keep going. We'll take a little break.
Soon after, I tell you a little bit more. Okay. So before the break, though.
Ha! Okay. Now I'll show you the cool algorithm. How do you go from some random values to this?
Right. You remember I draw this, I drew the fat lines. So the random value for that actual centroid was complete across my ass.
And the random clause, my ass. And the second random value.
And then do a third random value. I verbally describe it to you.
Then I can show the animation. So the algorithm works. You have to iterate many, many, many times.
Each time you do a little bit of what I'm going to tell you. Okay. Take the actual data points there.
All these are actually the real data, direct data points.
They'll never move, right? Because, you know, they're where the document words are.
So do the following for a different color. Check you for any one point.
Do what I'm telling you. But you're going to repeat the same thing for all the points for any one point, right?
Maybe that point right here.
Find the distance to the fake centroid initial gas finder distance the initial gas distance, The initial gas and distance.
The initial gas. Initial distance.
Initial gas. Find the closest of those initial fake centroid and assign that point grouping to that particular set.
So now what's going to happen is that point will actually be assigned to that centroid because that centroid appears to be close to my point.
Let's turn to the point this point over here again, find the distance to my fake centroid distance of centroid distance rec center.
And this will actually get assigned to that point to the fake centroid.
So when you're done with all of the points,
every single point will be partition and be assigned either to that fixed centroid or that friction that breaks into it.
Although that fiction drug so far, so good. Right. And now you're flip it which is for that fake centroid.
Suppose some points to assign to it. So suppose maybe only that point is assigned to perfect centroid.
Then here's what you do. You take the points that belong to that fake centroid.
All the points that belong to the fake centroid. And then find their sigma X like find all their x values and divide by and find a center for them.
And then likewise, take all the Y values and sigma y divide by.
And what that'll give you is for all those points, Right? It'll give you some kind of a centroid.
I'm going to use some kind of star symbol for the star symbol.
In other words, if those points actually are part of some cluster, that is where the centroid should be.
But no, the centroid was up here, right, Doing a little delta movement.
Make a vector between those and that, displace it by 0.01 like learning rate some small multiplication along that same direction.
Do this for that reason. Right? Then what'll happen is that centroid was here.
Right now we suddenly move to a slightly better location, so to speak.
This centroid will move to a slightly better location.
This centroid also moved to a slightly better location than figured what happened in the past.
Do the same thing over again. Meaning for the new three centroid, which got slightly moved.
Take every single point and find which centroid of the three is it closest to and assign it and make a new set
and find the center out of that move at a point so that centroid start moving when the center is still moving,
when the centroid moves away, then guess where at some point it got randomly misclassified would not properly get transferred to Cool, cool, cool.
Algorithm always works when I try it again. Okay, so then what?
Then how does it can reach It converges because you do this many, many, many times.
You watch the centroid move every time you know it, right?
It moves. They're all moving all the three at once.
Move, move, move. But at one point they all reach where they need to be.

One more iteration will not cause any change at all, meaning all the points already in a certain centroid, the one that's going to be no switching.
Everybody already belongs to where they are. That is when you stop. Okay.
Cool. Yeah. Then the time complexity. You can go read about this if you are okay.
It depends again on, you know, how many attributes, you know, how many points, how many clusters.
So attributes means how many dimensions, how many times you iterate,
and then how many points are in a dataset and how many clusters you want is simply a multiplication.
All of this complexity, you can think about why that is obviously because you're looping through all of them, basically.
Okay. All right. So we should do this pretty neat. Oh, well.
Okay. I'm going to request that. Hmm.
Wait a minute. Oh. It might be.
Oh, there it is. Oh, okay. So now it's just I did not scroll too much.
Okay, so what I just told you is actually over here, Tech.
You know, it's just Android's repeat form clusters. So for a centroid.
Ask who are your members? You know, form this right? Then for those clusters, find the mean for them and move them.
Sorry. Compute the centroid. Right. That just means displace the centroid along its opposite location by by a small delta.
You cannot be greedy and jump too much because then they'll actually make mistakes on center like in machine learning.
You cannot make the learning rate be too high for how much you want the error difference to being propagated.
Then we'll actually lose the minimum. Okay. The last functional will actually become like large arrogant.
Okay. See, that's all. It's exactly my idea.
And the cool thing is, finally, we will reach a point when the centers don't have to move because all of the points don't have to switch membership.
You are the lowest quarter greedy algorithm.
So, yeah, that's all it is. Everybody's vector, you know,
and you have centroid and that how you make the centroid right at all the x sigma x divided
by the count cluster centroid count and then likewise in know sigmoid whereby by centroid.
So then I'm going to show that to you finally. Look at that.
Well, it was time I had space. Okay. Watch this.
Okay. So obviously, these colored circles are actual data. They're on mobile cam.
That was the initial guess. And the guess exactly went to where it needed to be.
Here the guest was almost at the right place. It didn't have to move too much here.
The guess was here and even turn direction, made a little bend and went there here to get started here and then went here.
It always works. I'll do a few times. Watch it. Okay. And by the way, there's also a random number of clusters, meaning here there are four clusters.
There's a value between three and six or two. Okay, So play with it.
Now you have five clusters you're asking for and again, five movements.
Look at this one. This is all the way over here. Moved this one all the way over here.
Moved. This one did not have to move too much. This move very little.
Completely random initial guesses, but stabilizes and all converges every single time.
It'll converge. Every single time. This cannot fail.
Now I'm going to show you the code and we can actually, you know, maybe before the break, just for fun, modify the code.
Okay, then here the same thing. Wow. Look at that one. So this one started here and possibly initially had that one misclassified.
Right. But then it got pulled away and then that got classified over that one.
That's a pretty neat one. Look at how far it had to go. Okay, let's do it again.
So all random in JavaScript randomness. This one also initially was here.
So that point initially was misclassified. But then as soon as this went away, that point corrected progressively.
It never fails. Came in clustering. Cool.
So what we should do is look at the source here.
And at the very end. Oh. Oh, my God. Even smaller.

I'm sorry. Okay. I'm going to save this in just a second. See?
Look at the very end. We can actually ask for how many data points we want.
See? Give me between 20 and 60 overall documents.
How many clusters do I or do I want some random number? Between three and six.
We can hardwire that to five. Okay. So let's actually play with this. We can copy all of this.
Control C, go here and make a little fun.
Okay. This looks fine.
And so then all of what I told you is all here, by the way, you can actually go and look exactly what is happening.
There's one function called came in.
So can we just call that a day and see that it came in that that one new came in and just simply say, What's my data?
Meaning what points do I have and how many clusters do I need? Great.
So we can then save this save, but then we need to rename this and then call it like something that is HTML.
Right? Like a fun dart H html does tell us a dart text extension.
So I need to somehow going. What is a good way to do that?
Normally you need to show file extensions. Okay, but let's try it. Okay, so I'm going to go on desktop Finder is HTML somewhere.
It says some crazy thing about customize.
Does anybody quickly know? Properties.
Oh. Oh, I see.
And I actually know that there's still some type of wireless text, you know?
So what we need is show hidden extensions. Does anybody know where that is? There's a link also like hidden file extension, you know.
Okay. Oh, this one. There you go. My God.
Okay. Huh? All right, so now at least we got these HTML backs, and then we get a little rock core.
So now no edge. Come on. All right. Edge being the answer.
Use me. Use me. Now go your step. Now we can edit that and actually modify what I told you.
Let's make one. Let's make, I don't know, 500 points.
Right? Why the heck not? So points. 500.
Why not? Okay. And then let's make, I don't know, ten cent rates?
Probably. Right. Ten cent points. Okay. So we can save this.
Then go back and. Oh, my God. Okay, so that's one of the points.
And then ten Central talking to that. Those points went almost on a grid.
That's pretty funny. See, that is exactly where the random numbers help. Okay.
So I'm going to go back here and actually make it random, you know? Yeah.
Okay. So random. At least you can get very the number of pointless.
Let's make it random. I don't know, like 30 or something. Okay.
And then this will make it three and four. That only be three or four. Just to show you.
Fair. Now we can see.
Cool. Okay. So I did three. Four.
Four, usually. Okay, look at that. I mean, look at how simple this was.
So then that is the magic of. No, you know, this forever came into question.
It's a very important data mining algorithm, one of the best. But then you see how it works.
Start with an initial guess in your networks, in all of your networks.
These are all neuron weights, senior and weights, all completely random and true back propagation.
The weights become ideal weights and network learns in a linear regression.
These are all initially random slopes have a bunch of data points.
Right? And I want to say make me a regression line. Okay. You want that line you want with that slope and that y intercept.
But initially, both those numbers are random guess. It might be that, you know, intercept is wrong, slope wrong.
Then the exact same kind of an iteration happens.
It is slowly try to move the slope either away from it in which case the error bars to see these are the error bars that would become larger and more strongly when you move the slope the right way that it's going to get smaller and smaller.
It's like an error parabola almost. So I say this is the initial like y intercept.

Even if I make the y intercept even larger, say initially I was here.
That's my y intercept actually. Here, say say this is my initial y intercept and that's my initial error.
I make the y intercept even larger meaning. Suppose I go this way, then the error would actually go up.
That's what tells me. I need to make the y intercept go small. So go small, go small.
And the error would minimize. Same for the slope. Okay. Okay, so then it's all the same idea.
Then I can start with knowing nothing for iteration, magical iteration.
Learn it. You can learn regression, You can learn your networks. You can learn clustering.
Learn to learn. Hey, speaking of how we should take a break. 619 I'm just going to stop.
Okay, Let's take a five minute break, six minute break to make it even come back at 625.
Right. And then I'll go on to the next topic. A bunch of things that I want to tell you, but then you know, why the heck not, right?
I wanted to show you. Something that is interesting.
I meant to take up the break for this, Right?
But we had students, which is totally okay. So I'm going to take like more than a break for now, just two or 3 minutes.
Watch this. It's actually cool. I'm going to run this installer and then show you something fun and then we can get back to our business.
We're talking about classification now that we're done. Mostly clustering.
I still do a little bit of hierarchical agglomeration because no matter.
Okay. Okay. Oh.
Then can you please help move the camera? Yeah, that is true.
Hi, Dan, if I am riding on the board, can you please move the camera?
Yes, exactly. Thank you.
Whoa. Check this out.
Okay, look at this.
Actually, you know, I mean, even though we said Matt Tepper is actually going to make me do everything from scratch, which is actually great.
Okay. So fantastic. Oh. I might do a new song just to make it faster.
I'll actually do it yourself. Okay. This is a lot easier.
See? Check this out. I'm going to make a little pattern here. Okay?
This is called a step sequencer. A step sequencer is basically a timeline where at this particular time, only that instrument is on a disk is on here.
This. Check this out.
It's a topic that most in most of us.
Okay. Still loading.
One last deal here. Okay.
Cool, right? I got recorders out.
Okay, So we can just keep on going, like the scientists only go back here.
Yeah. All right.
A little bit of our music generation for you. Oh.
Oh. You know, try this stage light.
Okay? Definitely try it. Too much fun. Okay, so we have her.
Not that. Yeah. Hi there.
Yeah. So, you know, doing the break, one of the students had a question about how fast all of this kind of data can be searched,
you know, even even just for good old classification. In our document clustering, a new document comes in.
How do you know what cluster to go? Right?
You cannot possibly do, including distance or even L1 norm or cosine similarity with every single document that there is.
So what we do is we subdivide the space that's called a hierarchical indexing into D,
we subdivide all the space and make one big square and we say any single point at any point anywhere has to be in that square.
A starting point then re subdivided into four squares and say northwest,
northeast, southeast and southwest make four children binary tree squad, tree.
So then all of these points share right? You only need to go this way.

It means you don't have to go here at all.
It would always turn to for personal data and then for all the points within that recursively do for more subdivision.
Keep on going till every single point till a single existing point is like a leaf node.
That data structure is called a core tree in three D suppose add one more dimension.
It's got an arc tree. Now I can have a 50,000 dimension tree as well.
But same idea that Cuba is never going to be a 50,000 dimension cube.
That algorithm is called hierarchical small words algorithm.
So I wanted to show that in 3D, it's called orchestrate.
See that like a binary tree, but with eight children.
So you go down any one of them, but not to the other seven go down, one of them not rest.
And that is you rapidly get to where you need to be.
Okay. It's so fast because you throw away, in this case, 7/8 of the data in binary tree.
I think it's pretty amazing control footprint in our data.
Seven eight so the data your discard every single level and so that is now extend this in multi dimensions.
Then you have this thing called hierarchical networks, a hierarchy navigable in all small worlds.
So then you see descend through all the layers and rapidly get to your search is the same idea again and again.
Okay. But you can read this afterwards. So very powerful idea.
Same ideas in one layer, the topmost layer second later layer, the layers get more and more detail.
Okay. But that is how it can subdivide any number of dimensions. And then so rapidly get to if you look across here, you need to know that is going to be a possible result like instant, just in a couple of little descent.
Okay. It can get to where you are to throw away billions of not trillions of points.
Don't look at them. What a radical algorithm, 9 billion scale saying we're not kidding at all to that billion share similarity such.
Doesn't matter. Bring it on, you know, because we know how to do it.
All right. So then that's so cool. I hope you think all this is fun because you know and need.
Right? So easily Understandable. Okay, then I have like, a few things to tell you, and then we'll talk about classification.
All right. So short, you know, so it's a fair amount of time, what he means.
Question You know, if you'll understand.
And then I made this table copy, which is still over here, I can go and modify that anytime and then think of that one and then just run it again.
Okay? So we don't need to do that. And then the rest we already talked about.
So we use usually if you look in the code that I gave you, it is Euclidean, but you can modify that to be simple.
L One metric. L One metric once again is simply called I'll just say LP norm, I'll just say LP norm distance calculation.
Then it will say L1 l2 L3. LP space.
You can call this many things. See that that is L1 linear L2.
You can even have al3, L four. You can basically do anything you want. So there are like distance norms is what we call them selected.
Okay, That's what I told you. If this is true, then that is Euclidean.
If this is one that is the taxicab geometry, because you don't have to do a restaurant anymore, just simply remove that as well.
Just simply sigma excite. Meaning sigma expressing my y in our case.
Okay. X distance, right. Just count all the steps in either you know, X or Y.
Great. So then this just goes on and on like the same idea. A pretty neat idea.
All right. That's the classic L2 Norm. Okay, So then you can do on one less time the L2 versus L1 norm and you will see the difference.
Hopefully they can show you a taxicab. Manhattan Geometry, Manhattan Distance picture.
Oh, that one. Okay, This is what I try to draw on the board there.
So this is actually Euclidean, right?
But you can either do it this way, meaning at the red line position and or you can add all the blue lines or add L or any pattern.
All of them will give you the same answer. And then the answer would be difference between X plus distance between Y.
I have one request. No, please don't talk. Cooking or talking.

Please just be quiet, because it is very distracting to the students that are sitting in the class.
One of them came and told me in the break, you know, okay, so please try to be silent.
All right, Sell one alto cluster. Now, let me totally get that. Okay.
You can cluster data. It's pretty simple. Yes.
Okay. Means we went through this just now, and I only have to tell you, a hierarchical agglomerate of means start from individual point one,
point one, cluster membership one and go up like bubbles, soap bubbles.
Sometimes you blow like a lot of them and then the fuzed agglomerate and they get bigger and bigger and bigger.
It's like that. Okay, that is agglomeration or you do divisible.
And how do you know what to merge. I told you. Suppose you have like 11. here.
It's own little cluster right there. Here's another point is on little cluster.
Right here's another point.
You want merge this and this has one cluster to make a cluster of size two because those are far apart compared to this one.
Otherwise always only find nearest neighbors pretty close and make them into a slightly bigger cluster.
So initially they're all like one cluster, a circle around each one point.
Now suddenly you have two member clusters, two member clusters, only two points always.
Then you can have four member clusters when you cluster them together.
So you go up in clustering. That's why the top part is divisive.
Is a split always just split, split, split again, based on including distance Silbert approximately give you the same again merge.
I split one cluster at a time and again it's all based on centroid.
So ultimately, okay, so how do you actually know like, you know, like what is close and what is for other words,
how would you say how would you say that this cluster is run on the average, not the same as this cluster?
Like, how do you define this space? Okay, there's three different ways. One, you can do centroid calculation.
You can say that centroid and that centroid, you can use the distance to say they're far apart, so to speak.
Or you can use the closest point on here to here.
The closest point, that is one way to separate them.
Or you can find the farthest point between two clusters and use them as a metric to know what is farther than what.
So there are three different ways average in a max, basically. Okay, that's what that is.
So average, a single link center of gravity, you can basically theta.
But all this. Okay.
So like this, when you have initially all of the messages, their own cluster say these are documented on a cluster, various are clustering.
Initially everybody's in a one like a single membership cluster.
Look at how you use them as so far apart. Right? It should not be used by anybody else.
But look at BNC. BNC could be one unit, right? That is what this is doing.
Likewise, DNA should be one unit. DNS cannot be one unit because you can do DNA before you can do that is we decide success all by itself for now.
So then in my second level of hierarchy, I have a separate, then I have B.C,
then I have D, then I have, in other words, A, B, c, D, e, f, But I can keep on going.
I can say maybe I should combine DNS together and make a d e f.
Likewise, I can take the different b c together and make a BCD and then I can maybe add to it.
Then I'll get everything I'm going from small to larger. Okay. You can stop at any point.
If somebody says, Can you make me two clusters out of all of these, then maybe I should start with a and then BCD.
If somebody says, Can you give me three clusters, I should start with A, B, C, D, I can stop anywhere.
Okay. Depends on how many clusters you want. But then the ultimate limit is one cluster, which you probably don't want.
Okay, then that is where you go from small to large. You can also go from large and small, basically backwards.
Okay, the diagram know. But this one small difference between them though.
Okay. Also, regardless of how you do it, smart, large, large and small,
you can visualize them with a beautiful data visualization technique called a den diagram.
A den diagram, that one. It's not even a typo here. Is then program.
Oh. Oh, wow.

All right. You said I apologize, said Andrew. Sorry.
Same idea. So these are the ones that are being clustered. But initially, these formed one cluster.
Those formed the second cluster. Those clusters got merged.
And then this was these two are one cluster. This one all by itself got merged.
Those two got merged. Finally, when you go to the root of the whole hierarchy, that's like one giant cluster.
Now, otherwise I can show the clustering as a tree because there's a hierarchy.
I can show you like a tree. Pentagram. Great. Okay.
You can do it immaculately, by the way. MATLAB in this case, I guess.
Yeah. Did you know we have free access to MATLAB?
All of us are free MATLAB license, and MATLAB has incredible machine learning.
Tutorials. Between Elway.
Slash MATLAB underscore USC magic.
They give us so much to unlimited MATLAB, all of that stimulant availability.
But even more, the world told us there are so many see that you can you can basically do machine learning
like right 100% MATLAB can every single algorithm ever including clustering all here so much.
I salute you. But you can do it in Mad Libs as well.
Matt Plot Lib Dem program.
Ten lines of code. Take it. Learn Python Rough gallery.
StackOverflow. Oh. Me side by.
Without one. This one uses sci fi.
It's a pretty similar deal.
There's got to be a matlock live one somewhere, so copy again something so small in your Jupiter notebook and just run on CoLab Energy Drive.
It'll actually do it for you. It's like that. I actually plotted for you because your line start was right.
So little and a little bit more and then a tiny bit more.
Just do it. That function called us. So likewise, plot loop has a similar function called Saltzman already done for you to ship and then pretty
soon charge empty can actually do the call for you type which say give me make maiden
program and then it'll write this code and call my plot or in this case hyper because open
air has an API or you can do MATLAB calls so you don't even need the actual code to do this.
You can actually talk to it. Okay. It's not even funny if you no matter if you know MATLAB, it's useful.
But, you know, after a while, it's not going to be useful. Just know what you want.
That's really what it comes down. To know what's available and just how to ask for it in plain English.
That will set you apart from people that don't even know what it is. And then that's the end.
All right. So our hierarchy, you know, I mean, just not to talk about all this.
Yeah, there isn't clustering to start at the top again. Partition.
So again, look how you do it, though. You know, it can means you can take equal to two and then three, four.
Keep on going. So then finally it can get all the way down to one, basically going backwards.
The three that I showed you. I look at this. Bottom up methods make clustering decision based on local patterns, right?
Because it's only looking at its own neighborhood like a little greedy algorithm.
Because going from small to large account, it cannot have a global view of all the other points out there.
Whereas top to bottom is all it takes for everything S1 one and split set, right?
So this one is actually better. You get a better cluster a result when you do top down versus bottom up.
And I'm going to get this in. Paramount singing for a change.
Somebody singing Stop it. If you want to send comments in, health care should be cool.
Okay, so then that was that, right? I'm going to tell you more. So clustering is cool.
I understood clustering meant mainly the idea was you can use Cambridge clustering
or you can use bottom bottom to top or top to bottom three different algorithm.
Although the top to bottom also uses came. It's great.
Now that easily dovetails into classification.
So classification at 645. So I'm going to do the following stop here and.
Well, by the way, check this out to my point.
Okay. Somebody starts a graph plotting company. But secretly, what they do is Jeopardy, not API call.

Maybe, but then also from Jeopardy in culture, MATLAB, MATLAB.

So Jeopardy can call other things in the world. By the way, that is where the pure magic comes in the called plug ins.

Okay. Also this one generator axis.

Yeah. Do a little robot symphony connectors. I'm not sure if I told you this last time, but this Jeopardy store.

Right. So, you know, they announced the Jeopardy store idea, but Jeopardy Store.

You can chat with an alum and then you can customize the alarm with your expertise and then listed on the store and make money off of it.

So people are going to make all kinds of crazy. Basically go to Alexa into Alexa trigonometric functions,

but now you can make a chat version of that and then you can chat and it'll plot pictures for you.

And if you brought the question graph incorrectly until you you actually do the same curb accident things that Alexa cannot tell you.

Okay, Because that's all the speech is going to be like way more.

So, so much of knowledge people are going to do all of this.

Okay. And then make them into the story idea.

But that's all because all of them can be fine tuned, but fine tuning that on them.

Okay. So I say yes, the notion of classifying it's going to go somewhere with it.

Okay. So okay, so the idea would be let's talk about classification.

All right. Sorry. I lost my little train of thought.

That's what we were going to do. Get ready to Shenzhen.

Shenzhen. Shenzhen, Shenzhen, Shenzhen.

So, please, hopefully over here. Thank you. Perfect.

All the way. Hey, you want a piece of chalk? All right.

One more. Did Air Canada.

Here. They're reviewing the levee. Hopefully.

Hopefully. Check.

Wash, Check, wash. I to go?

I'm sorry if I. I'm so sorry. Okay. I don't know where this place.

See Silverman. Wow. What's up with all of you clustered over there?

Okay. Oh, my God. Okay. It's not an even distribution kick.

That's Kelly Canyon. So I think Kelly might be off, but it's okay.

Uh, Jay's hour. Yeah. Okay.

I'm going to keep calling till someone in the front says I'm here. You're all in the back so far.

Oh, my God. So I could actually see a teen.

Andy, please be in the front somewhere. Does anybody want to be in on?

They're get to the slides while teen and these are 18 and you're also in the back.

My goodness. So. Hum. Yeah. So we're. So I'm Mary.

So you're not here. Zion.

Okay. I'll send the back. Okay. I'll give up. Okay. Everybody's in the back.

It's an uneven distribution. Let's talk about classification. Classification follows naturally from clustering, assuming clustering has been done.

So we know what each cluster is called. So in a new document comes in how they classify that to be one of them.

Or when a new query comes in, the user wants to search for something.

What documents are we going to give them for the query part?

I'll give you a call algorithm called Roxio, which will actually make the query to be a better query.

You know, I'll just tell you, here's what it is. Supporting the words the user typed for the query was not the best word, meaning they could do even better, but they don't know the way sort of done it. But the query took them in their vetting space to hear.

So it took them to here. Here. Then you would do the following.

You would say, I'm going to serve the, you know, all these documents. Right?

But suppose the query had a a search radius because that's how the actual algorithms work, again, within a certain radius, within a certain distance.

They look for similarity, you know, like what? Ordering distance.

So then, because that query was initially centered at the edge of some cluster of documents,

if the query radius was pretty large, users going to get some irrelevant documents.
Also, because I didn't know what the search term meant, right?
They just type something, so they're going to get better answers. So instead of we can stop that from happening by taking the initial query that they
give you a right and then showing them a bunch of relevant and irrelevant documents,
meaning we know that this is what we should be giving them, right?
Because that point is close to all of them. But purposely, we also give them these documents as well.
I'm going to say of all of what we give you, like in other words, for some reason, do you want that you know why you type?
Then the user says, No, I don't like those documents. Then we can do the following.
We can actually migrate the query vector away from it,
away from the relevant documents that will put you right in the middle or away from the relevant documents.
It'll send you more towards the relevant documents. Then you actually satisfy the user's query.
And that's a better thing to do. That's what required us.
So Rock is all about taking that initial query and deflecting it towards a better point in the
embedded space so that any neighborhood that circled the retro is now suddenly centered here.
So now I have a better circle. So like that or that circle and that circle is 100% relevant use as well.
Just great. Ask the use of a feedback. Okay. And it's called again, relevance feedback.
Okay, I'm going to tell you that. Yeah. So then I'll go to the fact the first line is a matter of relevance feedback.
So how does this relate to tangibility, human feedback, reinforcement, learning that human feedback.
That. Yeah, see that?
Therefore, that's the relevant signal. In other words, some people were given they horrible things.
I don't want to go off another tangent, but I'll put it up on extra. Just today they released a free press release to prepare.
There was a competition that I think resulted in 600,000 prompts.
These problems are all jailbreaking prompts. All of them are able to fool the top three alarms.
Charge a pretty and there's one called flan. And then also the other one borrowed one, you know.
Okay, So then somebody compiled all of those and basically gave it away.
So data set, there's a dataset of almost like evil in the sense that these are all the ways in which you can break the top three engines.
Okay. And also, they made a paper with that. Okay. So yeah, I'll find it another time.
But meanwhile, this actually works somewhat. Okay. I can see that human augmented text.
Okay. So that is the whole idea. So in here, you can make the classification be better, meaning serve them better results.
If you're actually able to take their prompt and then move it out like a better area.
Okay. So centroid already. All right. So, again, Sigma X divided by the number of points.
Sigma Y divided the number of points. Just little review for you.
All right. So you know what the centroid is. We don't need to know the words X plus x plus x plus plus plus x divided by one, two, three, four, six,
eight Sigma white at all the Y heights and divide by eight and you're going to get to average height.
Likewise, for these blue points, these are diamond points and then for the x points.
So we know where the centroid is. Okay. Okay. So centroid then lead to a linear boundary between each pair of them.
That's where the convex polygon is called, where no is going to come up Again, I'll cut to the chase and tell you what the point of this lecture is.
Okay. Clustering already happened.
So clustering already happened on the cluster centroid really means if the clusters are far apart.
The centroid are is centric because, you know I mean I drew a circle so in three D is going to be a sphere in 30,000 dimensions.
It's going to be a 50,000 dimensional sphere, but it's still spherical.
It's not any of the weight ship. So that is ideal. But in the real world, all these figures are like real large.
They start to overlap with each other and that is when the spheres will basically squish muffins.
Okay, in the muffin tin, when they make muffins in an open plate, you have like all these little muffin little balls, You're making a victim, right?
They're all accusing each other of controlling the picture muffins.

Accusing each other. That's a boundary between two muffins if you're going straight line.
And it's called overnight Polygon.
So therefore, ideally, the order boundary is going to be spherical, but in reality, the boundary is going to be like a linear boundary.
Okay, so then that is really what you need to keep in mind.
So when actual documents are served, what actually happens is when your query point lands somewhere, you know what is called nearest neighbor stretch.
In other words, you ask the query point.
So the quarter point is where the question is, you say is the query point closest to the centroid or the centroid or the centroid,
and they go with the closest central, you know, one nearest neighbor search.
Okay. So that's one way to do all of this. And then using the word polygon, basically what you're asking is in what Warren I region is my.
Quite a point in because unlike the spheres writes the empty space between them.
Right. But overnight, polygons will have no empty space at all.
All of the two dimensional plane has been partitioned by these convex polygons.
It means unless you're at the edge, that it'll look like a case that only in case you will actually be inside the polygon most of the time.
Okay, One polygon, then that's like one nearest neighbor.
Because if you're inside one polygon, you know, like this, suppose there's a polygon here that supports the second polygon,
but one like that and a third polygon that went off like this and then a fourth polygon got a little weird.
Okay, They've got a fourth polygon. Suppose you're here. That means what that means is you are close to the center, right.
Otherwise would have been somewhere else. Meaning? Say you are here. Then that means you're close to that center wherever you are going.
Some polygon. That indirectly means you are close to one center.
You can always use a one centroid as one nearest neighbor. The nearest neighbor classification.
You can do it that way. And k nearest neighbors is called a K nearest neighbors algorithm.
So K can be one in which case it's called overnight. Or we can leave K to be not one.
We can make K to be three or five and then make an odd number of them.
What that means is you want then the whole of our nothing is thrown away.
Then what you do is say the K point was over here over the course of this.
Then you add not obviously thing that is a circle.
Instead, from this point you will do nearest neighbor search tree, nearest neighbors.
Okay. K is going to be three or 5% and over three years neighbors decide do avoiding like what majority neighbors are getting.
In this case, I'll get two diamonds, one circle. Right.
I'll go with the diamond majority. So the choice for K will suddenly change your classification of the words.
The user asks something. You'll start showing them diamond documents.
I suppose the circle documents. Okay, so K is equal to one, this one in a polygon case.
So they can either be one and not one doesn't break. Okay, then let me show you all that.
Yes. So again, this is all about drugs to Rockville Center.
Rockville. You already know what is what I learned, what is irrelevant. Meaning if the query went here, then obviously these are relevant documents.
Anything else is irrelevant. Documents again. Then imagine. Then you call this not enough relevant documents irrelevant.
Then you want to maximize the difference between these similarities. In other words, say the query point.
Q was actually over here and query point was here. Then this is centroid for the relevant documents and this is centroid for the irrelevant documents.
So that is the distance you want to maximize,
meaning you want to then say that that query is going to be served by these documents rather than these documents, because are queries pretty close dissimilarities higher than that similarity of other words.

This point, this is shorter than that for the shortest distance quote.
Okay. Then again, ultimately, all of this is done with the cosine similarity anyway.
So in other words, ultimately, you know, when you say like what? Why is this you and close to this?
Because ultimately you are doing cross and similarity. The cosine between like this and this is shorter than the cosine between this and that point.
So it's all like going in circles, but then you get the idea. Okay. All right.
So then tell you more about Rock Hill Hard. And this is the whole magic deal.
Here's where you take your actual query point, your credit point to secure this cute quick point you.
And that's the actual vector. Okay. In other words, say sorry about the massive picture.
I'm going to keep using this. That's my quick distortion right here, thankfully, to not get messed up.
That's my actual query. We're going to call Q. Q But I want.
So then that is correct there. Rather than accept the Q as it is.
I want to turn to Q and. Q Prime, where does the Q Prime I'm going to move.
Q to possibly this direction and call that new thing.
Q. Q. Q Okay. Yep. And that is done because you take the initial quarter vector and the article little thing and look at what you're trying to add.
You're trying to add like where the relevant centroid is. That's a relevant centroid.
DJR Divided by D.R. That is simple. Literally that point. That's that point.
Likewise, this here D.J. or at the end, that is going to be some relevant centroid like over there.
So then that is subtraction. So you want to find that gap between them, just about this vector subtraction, whether that means the gap between them.
Okay. That gap between them is what you want to add and make it go in this other direction.
I'll show you a picture pretty soon and you'll also wait. Them will have these alpha, beta, gamma.
And then just say, Yeah, if the user wanted some kind of query, I'm going to wait on query pretty high.
I'll say cold one if you want to show them relevant documents.
You want to make it better to be zero points on your relevant documents can also have a value,
and that's a pretty small value these days in many algorithms.
We actually set gamma to be zero, so Gamma zero.
What that means is we don't care about irrelevant documents. Okay.
That's a little note you can make if you want, but the numbers you can actually play with.
Really? Okay. So then again, I'm going to show you this nice picture.
The user made some query and the query actually went this little red triangle.
But look at the neighborhood. It is not all purely one x circle.
Right. So then if you show them based on some near neighbors, they might get something irrelevant.
Like they might get circles, but also they'll get access. So how can we move this away?
Okay, look what happened after you move the vector. That's my delta.
In my diagram, Delta went this way. But here, the new query went somewhere here.
But now look at the neighborhood. They're only going to get circles. We actually made the query better.
But how did we make the query better? We asked them in this area, like, what do you like?
Like, do like circles do like access. You give them a better board and say, what do you choose?
Then they're say, you know, we like I like all of these. I don't like all of these.
And that's what's going to determine moving away from all the X's, okay,
Because actually then becomes not relevant, becomes DENR versus circle will become D are circles relevant?
Basically, the user tells you in what direction the query should be modified.
Okay, so that's all of this is, you know, so please, you can read this by ourself.
I've told you some of them. And if I didn't tell you, don't worry about.
But in the exam as an example. So then again, imagine this like one class, another class.
Okay, maybe relevant, irrelevant.
If the user square falls somewhere over here, it's going to be in the bad neighborhood because it might possibly give the user one of these.
You are in a move the way all the way to maybe over here, in which case would only get relevant grade.
That is a toy example we're going to use to illustrate this. Okay. So far, the user did not say anything.
Okay? We just have to class our documents, but then the user comes in, makes it clear the query that the user makes.

Unfortunately found here this user squared. The parameter is even though it seems to be nicely centered here, that can also include,
unfortunately, some documents that they probably don't want to want to move this away this way.
Okay. Okay. So then actually, sorry, that's not the user square hypothesize.
That is actually the center of this that centroid likewise under this. And then comes the query afterwards.
Sorry. Okay, then what I'm showing you here is just simply the centric.
This pretty simple.
My question is going to be under the centroid of irrelevant documents like this and this going to show and then the query comes next.
Okay.
So then but that if you're just simply go with the existing centroid of the document itself to serve the user, then if you draw some kind of a circle,
not very small radius, but some reasonable radius, that centroid will actually include irrelevant documents also like in this case down.
So maybe it might be better to not use the actual centroid as a place to serve documents if this is what the user wants.
You want to move this way. Conversely, if the user wants more access, you want to move that centroid across here.
So it'll be, you know, more towards the right hand side depending on what the user once again.
But right now that centroid is not the percent rate. So how do you move it. Aha.
Okay then like I said to you, these are just basically complementary complements, but we're going to do something new with what are these?
What are we going to do? We Well, okay, so once again, same thing.
Sorry. Yeah. So this is one of them. This another one.
Okay, so far, so good. Now comes the new part. Okay, so take those two centroid.
Right. And do something very interesting. Find the difference. Right.
And then added to this way. See that there is a difference between centrists that distance between them.
He would use that as a vector by the vector difference. Ah, the vector difference to the centroid, but in the other direction.
So basically push to push that point the new location of the centroid.
Again, remember why you care what centroid is Because you draw a circle and then serve users at fault documents that fall within some kind of radius.
So you don't want to be where this is. You want to be away from it. Okay, then use that difference to move away.
It's pretty simple, right? Just in fact, Yeah, that's actually pointing that way.
Just go that way. Well, that's it. So take that direct, literally that vector, displace it.
In other words, this land direction is the exact same as this land and direction.
So now use this new point. Cool. So when you use the new point, this new optimum point, when I draw a circle, a circle only include circles.
In other words, this such search radius circle lonely includes this little circle.
And that's better than being here.
Once again, the summary is I don't want to be here because if I draw a big circle, it is going to include some axis.
Possibly. I want to move away from axis. So then what is the best way to access the particle repulsion?
Okay, more for. Suppose. Suppose you smell your body, order your bow intuitively.
What do you think are the step back? Right? Get away from me. You go take a shower.
I in the opposite direction. That's all this is doing. I mean, you probably should not go towards that, Right?
Okay, then that is. Then the rest is simply pure detail, you guys.
So now when I draw a circle centered at that new location as opposed to this location,
you basically said the circle, the circle, we strayed from this being the center to that being the center.
In other words, we went some.
When from that I should draw a circle that Prewitt went from that that circle to this new circle that I'm showing you on the top.
Okay. So Raqqa is very basic radio. Okay. So it's called Rock Hill.
You know, our term for a relevant case answering because the user tells you what is relevant.
Great. Yes. Then.
Right. Okay. So then in reality, though, because these points are not that separated.

This one with a circular boundary, it's going to become a straight line boundary.
And that is the whole of our entire polygon deal. I'm going to show Muffin Tin.
My friends. My friends, I guess. Hmm.
You can say a tray of muffins actually, is what I search for,
because it tends to actually make them separate and essentially not what I want to show you.
I just want to say a big muffins in a tray. Okay. So bake muffins.
And then you'll see the things that I'm talking about in a tray.
Almost. Not those kinds of trees. Like a flat tree, I should say, in a pan.
Oh, my gosh. Okay. That's not what you're saying. Uh huh.
I'm just trying to look for something where it is fuzed together. All right. That's really all I want to do.
Almost. Look, I'm going to go for chocolate chip cookies.
Cookies in the pen. We will get there.
Are. L must give up something like this.
Okay? I mean, it's not quite what I want. It doesn't matter. The idea would be that they're all like, Ray's okay with me.
Mm hmm. Me.
Hmm. Bread bite's like little donut bites.
I'm looking for all kinds of crazy things. Okay. Ha! Finally found one.
Oh, my God. Okay, so look at that one. So ultimately what I'm saying, normally the rockier distances would be like circles like that, right?
But when the circle start to expand, when the documents get more and more, less defined,
they become more and more like mixed together and they reach right up to a boundary.
And that boundary would actually look like that. See that? That became a straight line.
That became a straight line. Straight line, Straight line. You see that fuzing together because that expanding, right?
Just as like water in a polygon. Okay.
So then that is ultimately what this is saying, the connection between that and this line that said so not a circle, but a hydroplane.
It's called a hyper plane because in two dimensions it's a line in 3D volume like this,
it's actually a plane like, you know, this this little thing is a plane.
Cyber flame in 3D, two emission plane in 4D 3D plane in any dimensions, it might be an minus one dimensional plane.
If we call them hyper plane in 2D saline. Okay. All right.
So that's all. Therefore, then you have these lines and then when you are given these multiple centroid that basically are in between those lines,
basically talk about learning polygons, then any inequality that is coming in will be in one polygon,
but not the other polygon, second nearest neighbor assignment. Okay. That's all I'm going to say to you right there.
See the rocky road rocker classifier I told you about two different rocky years, by the way.
So oneness, the notion of rocky relevance, feedback that the rocky road is unfortunately called Rocky, but it's got a very different meaning.
This is simply more about rocky or classification.
You shouldn't even call it rocky classification, call it like a single nearest neighbor classification that's similar to Rocky.
So don't be confused. There's no relevance feedback, nothing. All we're saying is we have no polygon boundaries.
Then any query point will fall within some overnight polygon boundary.
Whatever polygon, the reforms in that polygon name, the cluster name is what you want.
Okay, great. That's what this one centroid. Okay.
All right. So again, this goes back to the previous slides.
We're going in circles now, which is classification, you know, but before that you need a cluster cluster, then classify, cluster, classify.
There's nothing new here, by the way. And I told you all of this in the beginning of the class, so nothing new, actually.
Great. Then this again, was about how what are different ways to classify?
Okay, so manual classification completely failed. Okay. This this was allowed in 2017 and finally they shut down Dimas the Amazon dot org.
So demos, open directory. Just no way it's going to work. Okay.
I have no idea how they even went up to 2017. So they must shut down.
Shut down? Yep. Incredible that they, you know, went like that far.

But if you look at the Mars, you will actually see what I mean. It's a manual classification effort is basically like Yahoo! Open source. Okay, Yahoo! Gave up. And then these people, my side will take it over but never went anywhere.
So look at that. Imagine going to do a search.
But now, I mean, you do some searching, but the search will still take you only to previously defined one at one category.
There's not an open indexed based search at all. Basically a classifier search.
Yeah. Then that is not scalable. That's what it says. All of them fail.
It is great when you have a small bunch of scientific documents, You know, some expert can classify it,
but as soon as it becomes quite large, billion, trillion Web pages dynamically generate Web pages, all completely fails.
Okay, so Yahoo failed. Google succeeded, basically. All right.
So how do you classify it? Very simple. Again, in also some instance, which is basically, you know, some kind of an element,
a new document is being asked to be classified into one of multiple little classes or labels.
You know, those all of this is just a total number of documents.
All the documents belong in these distinct categories. In this case, all the documents belong in three categories.
So, you know, C is equal to C once you do common city. Okay, great.
Then ultimately, you want to find out from some new document that is coming in. So it's C, are you C, one, C, 23.
That is all so simple. And so then that's a classifier function.
So the rock rock your classifier is, say, using water.
No polygons to say, I'll find the one nearest centroid that my document is going to fall in the way and show you a picture very soon.
So this then talks about the whole vector space embedding idea. Again, like I said, it's one of the biggest ideas in the whole generative business.
That is what makes music generation possible. Everything possible. But it's an old idea.
Turn everything into an embedding. Everything turns into an embedding, including document words, keywords.
So once they become embedded in this vector space, they become multidimensional data that, again, you know, if you want to look at a picture,
I can show you, I'll just say vector embedding and I'm going to say similarity sexual explicit because of chargeability all that, right?
It became like a very big deal in the world, but it's not been a big deal the entire time.
C Like, right, okay, It's been there with us for a long, long time, but now suddenly it's like pretty big, say, like a truly embedding model.
So this to wonder can take music or sentences, a video or anything and turn them into all this multidimensional dark.
So once you have multidimensional that you can do a similarity search because your query
is also gone through the same embedding and then went into become become a new dot.
And then the new dot asks what is around me? First of all, this what embedding is?
I told you that all just simply Python lists that one point in like in this case, say this.
I say that there's 20,000 of these, then you have 20,000 dimensional space and that is one point.
And the one access the value, 0.6, second axis value 0.3 third, actually 0.14 taxes.
What what So many dimensions. That became one point somewhere.
This became a second point or 2.4. First point. You can have billions and billions of points still like all of that right next to all that,
all the drone stuff, helicopter, we look at this Goose Eagle, B, all the birds and insects went this way.
Helicopter, drone, rocket, missile, jet, plane, all went here.
Obviously, that's where similarity comes in. So ultimately, this is the whole I mean, it's a way to summarize everything.
Words, sentences more correctly, even become vectors.
So there's something called tokens, right? Token. It can be one word, token can be a whole sentence.
You can embed individual words, you can member also sentence.
So sentence transformer to start even looking for things like this. Okay, sentence transformer.
Then you'll see a sentence. Former.
That can embed entire sentences. Okay. Yeah.
Okay. I've come back several token token in this case, token as a sentence.
So again, the same idea, see that you take a whole sentence and run it through embedding system like a neural

network,
and that will become some kind of an embedding in a multidimensional space into sentences one at a time.
Or they can do words or music or images and better on that.
Okay, so then similarity such. Quote.
Again. Same idea. See all this? Okay. Those are the natural boundaries.
Okay, Between things. That is ultimately what we're saying.
So in some 100,000 dimension, your kwe is going to look around and of course, similarity or even including similarity, L1 similarity and return.
Obviously things that are pretty similar to it, but not things that are pretty far apart. Okay, so what is the most efficient way to do it?
That is what all this about. Pine cone is one vector database that just nothing, just vectors.
Whereas you an oracle these days, relational database can store vectors, so everybody can store vectors.
But I want to get back to all this in the last one. Look at this.
Even between books, crime novels all go here because Rockport, murder, blood, the police, fingerprints.
Right.
Whereas children's novel talks about very innocent things like apples, birds, butterflies, ducks, they go somewhere else, and then you can classify.
It's pretty amazing. Yeah. So then, you know, they all become like vectors.
I showed you so many examples that all each is a vector vector because it's a point.
But when enjoying the point to the origin, become the vector vector in point A basically, you know, they're the same units.
Okay, but mathematically are distinct objects. A point disappoint vectors vector.
Yeah. Yeah, it must be proprietary.
And exactly. All the actors in our store are the same. Exactly.
You know, the actors themselves might be the same embedding model or give you the same elements,
but the way they store an index, they might actually be highly different. That is where Pinecone Chroma, we create all that.
They're actually better because they're from ground up written for this whole and are indexing this way,
whereas P.G. Vector never Neil Folger can actually do vectors these days.
Yeah, they're basically trying to retrofit on top of the existing knowledge graph in our relational databases vector format.
So I'm bit on scale as much as a problem to a small number of data.
You can get away with it. But at some point I think there's a performance difference that people say use a pure vector database.
Yeah, good question. All right, cool. So, you know, then the same idea, can you have like one class are all similar and other classes similar,
but between the two classes, they're very different. Moving on.
Okay, so same thing you already saw before. You know, this, like, repeat.
Okay. That's why this can go so fast. Okay. By the way, when you generalize this sometimes called a Minkowski distance in Russian mathematician.
Okay, Where P can be anything big can even be, you know, like, for example, three, four, three whole flat space, right?
And then like, hyperbolic space, negative a career, reciprocal space, positive curvature, space, curvature is all determined by what?
Well, for curious. Okay, I can really show that you say hyperbolic space.
Hyperbolic versus spherical space so math can let you deal with.
So a distant distance between two points in a flat space.
It is Euclidean on a curved space, positive or negative. We lose like on the surface when earth or on a hyperbolic plane.
So hyperbolic geometry versus spatial geometry going on this fast here.
Okay. See that? That is what a hyperbolic plane looks like.
Ashmore From the center this distance, by the way, is equal to that distance, believe it or not.
But in Euclidean space, they look big. They look small, right?
But that is the little anchorage you see here that is negative curvature, hyperbolic Euclidean, zero curvature, elliptic positive curvature.
And they all just simply depend on that value for Q And in this case, cubicle two, less than two.

More than two, Less than 2%. More than one. Very interesting, right?
Okay. And these you see nature all over the place. Cauliflowers and mushrooms.
And I've got this critters going on. All right. You can do knitting and crochet with hyperbolic geometry.
It's pretty fascinating. That's why this is so in. It's too flat space Euclidean.
Okay. All right, so then, yeah, assume this already done, Right?
Already done. Classification. Now, the idea is when a new document comes in that's manufacturer document, like, where should we put this?
What should we call this? You look at and first of all, your total event documents.
So for the event, the new decay vector, we call it a quarter vector comes in.
Look for the k near neighbors can inadvertently k can be tricky, can be five case and odd number Y is cannot number.
You know, because you want a majority, you want one to be more than the case even then.
Imagine that document the query vector one here. Okay.
Then if k's equal to four, then what might happen is in your search radius, two might be one, class, two might be another class, and you're stuck.
You cannot classify it. So always go for hard numbers. That will be two plus one, one plus two, you know, then it'll break the time.
Okay. That is all it is. So pretty simple. And when K call the one that's called nearest neighbor, then that's where in a polygon.
Now you know everything. Okay. 75 neighbors. So then now case file, right?
Supposing you already done clustering. So, you know, this is one cluster government documents,
one master central documents on register all writes this acquiring point a new documents, somebody upload a PDF file.
What should we target the speed of automatically document classification You know
let's look for compared to where the document is for five nearest neighbors.
One, two, three, four, five. Thankfully, of the five it is four +14 of one kind, only one of the other kind.
Sorry. Since your last welcome to our government document collection that is also in this case case file K Quest three again,
it's all the Mediterranean's neighbors. One two.
I'm going to be generous and go that okay not just so that that's still to propose
to my centers and one green so I still go to magenta 11 go with one nearest
neighbor should be alternate polygon this still wins so on this neighbor means
I'll show you pretty soon there's going to be invisible lines between them, the convex polygons.
Okay. Otherwise they remain curved like this. Cool.
So no matter what you do in this case, by the way, you in take all the one. Yeah.
This just like a side trip cable.
A six foot tall order, huh?
Six nearest neighbors. Yeah, I think this is trying to make a point.
Pretty soon they'll tell you, don't make that an even number in this case.
I'm sorry, a case six. But you still got away with that case. Should not be six.
Okay. Because if care was six, then supposing you're over here or something, Right?
So I over here, then it might turn out that there might be six of them in your circle, but three might be one crime, three might be the crime.
Then you can classify. It has to be an odd number. I think they're trying to make a point.
Okay, so let's look at these are numbers. Okay. Okay. Equal to one simple nearest neighbor.
But by the way, look at how it changes when k equal to one.
You get one square, right? Then this number A for the three.
It means do 1 to 3 nearest neighbors. Now, two triangles, one blue.
Suddenly a switch to a different classification. So it is sensitive to what the values are.
Case. And then on case seven, then of the seven nearest neighbors, four are here.
Three. Good. Four. So either one or seven seems to give you a square and three gives you a triangle.
So a bit scary, right? Meaning you would think the same no matter what. But the switch return.
Yeah. So don't make it to be too small.
That's a problem with only care to be able to one single point freely in which you're going with one nearest neighbor.
So the point of all of this is don't make care one that's too small.

Don't make two surrenders too large. Instead, go with three.
So this is probably like a triangle document, even though it looks so much like it's a square document.
Okay. But on the average, you know how to enhance neighbors, compare to one.
And then you got that right. See, this is exactly contradicting what happened here.
Right. So don't do that. In software development, what instead of called, you write a piece of code.
And then you say, don't write code this way. It's a name for that.
So. Yeah. Perfect. Echo.
Yeah. Samantha Pattern. Exactly. So that would be an anti pattern example of cake or the sixth kitchen and kitchen and weaving.
Okay, now let's look for the one. Then we can move on to our next topic. You're not going so fast, right?
So cool. All right. Yeah. Then the whole nearest neighbor business, you know?
Okay, so this is not Camerons clustering honest capitalism in Cameron's clustering.
It's a clustering algorithm. We already went through that. Now, what I'm talking about, this is K nearest neighbors classification.
We're not just a classification like a machine learning classification.
So there are so many classifiers, SVM support, vector machine, so binary classifier always.
RB It is a hyper plane and same type of plane. Or you're on this side or on that side.
Logistic regression usually is a binary classifier, but we can make it be like a multiclass if you want.
Neural network can classify how many of a label you want. Okay. Okay. So classifiers can be binary or non-binary.
So in the K nearest neighbors, what you're doing is again classification to multiclass.
In other words, in this case you can be one of three different classes. Okay. So the point is there are many classifier algorithms.
In fact, you can even look at different of all of them. Some of them are lazy.
So data mining classification algorithms, suddenly you will see all the things I told you.
SVM. You know, all of those are decision. Our decision tree is also a classifier.
So there are so many classification algorithms. See,
look at that decision tree native base SVM can and there was always there are so many right but of all of them
we call the K nearest neighbors a lazy classifier so lazy because in machine learning you learn something,
right? So in this case you a reach and you learn how to classify problem can and does not learn anything is lazy,
just sits there and does nothing until the query point arrives.
When the query point gets to me, then suddenly I can find and nearest neighbors, right?
3 million neighbors and sort them and say, Here you go.
So it does not actually learn to classify anything till the query till a new unknown data comes in.
So we call it lazy learner, too lazy learning algorithm. Cannon, lazy learner.
So that's the point we're making here. Lacy Lerner Because it doesn't actually learn.
Yeah. Lazy learning and this stores training data and waits until it's given a test.
I mean, right there, it literally just waits. Nothing to do. I know where the data points are.
Give me your data point. I'll tell you what. What I'm close to at runtime.
Right. So there's no there's no learning then, actually. Okay.
That's all it is. As opposed to eager learning. This one is so eager. I don't want to fail my neurons, okay?
I want to learn. I want to learn. So that's opposite of lazy learning. So this one is a lazy learner.
All right. So I talked about all this and. Right.
This one just stored the label training examples and at one time classified the left at the last part.
That's why it's called lazy. Yes. Then under a one nearest neighbor center.
One nearest neighbor. This one in a polygon.
So in one less neighbor on that closest example, you know, it doesn't matter how many around your picture.
You suppose you have a bunch of points already in view and they can be in whatever category, maybe which one category.
Second category, you know, all that doesn't matter to this learner.
You're some test point, meaning a new document arrives and say, What should I classify in your document as A or B, in this case binary to label.
It'll do a distance calculation within some radius. By the way, it's lazy.
So that's the closest distance, maybe second, closest distance, maybe third, closest distance.

It'll go with the closest distance. One one. Just one of them.
It'll be whatever this class is. Okay? If I done three nearest neighbors, it'll still be this class because two of them are described.
One of them are that kind. That's all. But when you do one with one nearest neighbor, that is same as Warren I polygon.
I've said the word polygon so many times and I'm going actually straight to. It's a pretty incredible spatial data structure.
That's really what it is. Okay. Over an Apollo conversion diagram, it's also in multiple dimensions.
3D 45 D. But usually 3-D crystals, brittle fracture happens in crystals.
Where the fracture happens along these these crystal boundaries.
If you drop a coffee cup, it shatters. And the shattering is because the cracks go this way.
No crack would ever go through the material. Okay. Yeah. If that happens, it's called tactile fracture is different.
Okay, So in this case, it's going to go to. Exactly. Because it's much easier to propagate cracks through all the boundaries.
Anyway, this what I'm going to tell you pretty soon. So now imagine these are documents, okay, this one kind of document.
So like, for example, biology, chemistry.
So biochemistry documents would be right at the edge here, but still biology, chemistry.
So then any query point in the very worst case, we are lucky the quality point might be right.
And that little vertex then you can classify that depending on is tree three or four slightly less lucky.
The quarry point might actually be literally on the edge. Then it can classify either.
But usually in more cases, most of the cases, the greater point is going to be inside, right.
I mean, the polygon, then whatever the centroid for the polygon, it's going to be the class.
It's like a single nearest neighbor classifier. Okay. I mean, it's a great data structure.
That's so I'm going to tell you. And her first name.
That's crazy. Why? Oh.
That's crazy. 2002.
What the [INAUDIBLE]? I was at DreamWorks at the time, 21 hysterical.
I wrote a paper on this about using one polygons to crack stuff in Kung Fu Panda, Thailand, of course, and destroys the temple at the very end.
And big rocks. My software is able to make those rocks.
There's no polygons in 3-D. Okay. Then it's in a polyhedra.
Okay, So then, in other words, I love this, this texture. The idea is in Kenya's neighbors.
You already know, right, that somehow there's all these polygons.
In other words, you think that somehow all the documents, you know, are clustered in a certain way so that there's a boundary between them?
It's all not completely fuzzy. If this polygon is mixed, say these are intertwined together, it's actually possible.
You know, it can make the data to be much, much, much more non linearly separable than machine learning can still do it, but you still know.
In other words, say take a paint program in Photoshop and blend all this like crazy.
I still know where the purple is. I said Nobody yellow is right, but it's not easy to draw any kind of line anymore.
Your networks are Deep Learning Network can still learn exactly what the boundaries are, can identify the simpler cases.
We don't need neural networks with all 20 OC So let's all add this to whole learning the lazy part.
All right? So lazy case based memory. Basically they'll mean the same thing again as opposed to eager.
So what is K? Yeah. Case usually a very small number, usually three.
Okay, that's all. And again that if that if the documents are not properly clustered of the clustering look all over
the place it's called noisy like some of the blue in this way it start not be a polygon anymore.
Okay then all this classifiers are completely failed because they all depend on
the fact that you can draw a line between any two regions and separate them.
That's all. Again, not to go off on a very different tangent, but look at this.
If I said TensorFlow, TensorFlow TensorFlow JS actually TensorFlow playground.
I'll show you what I mean by crazy data. Not necessarily noisy, but difficult to classify data you see here.
I see that that's somewhat easy, correct? Because, you know, there's 42 different classes, but easy to classify.
This one is a little bit difficult, but not too bad because you can draw the boundary right here.

Right? This one is pretty easy. There's almost like clustering. Look at this one.
Much, much, much harder to classify. How do you draw a straight line between them and separate them into a cluster?
You tell me. Not possible, but with neurons, we can actually do it.
Okay. With enough neurons. With enough neurons with enough even that I can say do it, it'll actually run.
And there is being minimize that to error minimization, I told you. And then it'll actually do it after a while.
Okay. And if it doesn't do it, you can add more neurons and try again.
That's the whole deep learning part. Okay, Can I add more neurons?
You add more layers at some point. It actually will do it though. You can play with this afterwards as opposed to something much simpler.
We don't need so many neurons at all. Okay, that's so simple.
But regardless, you are neutral. Only if you play with it, it'll actually work.
You can actually do what's called pruning in pruning and delete some of these neurons and layers.
And it still works. You know, then you can run an age device. Okay.
Cool. So then let's get back to our inner polygons.
Right. So then they have to know something about the data, has to be a little bit trial and error.
Okay. And then the cross-validation. Yeah. That's all about how to find the optimal value of K.
It's no big deal, right? Just skip it and just say three, then usually three or five heuristics.
Okay. Ha! Finally, the one in a diagram.
Okay, this way. Cool. Okay, So can you ask neighbors on cake wall?
The one I told you. That means you go with one near centroid.
It means you're always your point is going to be in some polygon versus some other polygon.
It's a space partitioning algorithm. Select that. It means for any point.
Right. For any point at all. For any point X is going to be one.
Right. Which is exile. It's going to be one. I says that the distance between that point and that centroid, which is what this is,
is going to be smaller than the distance between that point and any other centroid,
which is actually J So, I mean it means you're inside Polygon called I JS Any other polygon it'll always be like that.
You are almost at the boundary, almost at the boundary.
If I'm on the left side of the boundary, I'm still close to E compared to C, As soon as I cross over on the right hand side, suddenly I'm going to switch my allegiance and I'm going to be close to only on the exact boundary.
I'll be equal to both of them. In fact, that is how the algorithm works.
The way you do this algorithm is actually pretty amazing when you are given a bunch of points How to even draw the line on a polygon, right?
You would actually triangulate first. Should I do?
Race. Oh, my God.
So they should come in. I'll take the small front.
Check this out. If we're given a bunch of points like these and they say draw DuVernay polygon diagram for them.
They're not exactly some perfect lattice or something. You would do something called the linear triangulation.
The first triangulated first easy way to do it. Triangulate is an art.
We're not polygons, okay? They're just triangles, right? But they're a very special kind of triangle.
The special, because in this case, you know, these four points, right, is four points.
There are two ways to triangulate them. One way I drew the other way this way.
Right. The DeLong algorithm will swap that edge and not make it be like that, because that leads to this big, obtuse angle, right?
We don't want that. You want to then instead swap the diagonal for this diagonal and end up with more equilateral triangles.
It's a pretty neat algorithm. Delanie Okay, so let's do. Delanie The.
Delanie The reason to do Delanie is because from here incredibly we can get overnight
polygon how perpendicular by sectors for each triangle be that triangle for each edge.
Draw the 90 degrees perpendicular this edge also draw the same perpendicular like that.
This one was never going to do well exactly meet at a point in the middle where is that point called centroid?
Exactly. Very nice. Yeah. Centered. Likewise. This one also is going to be a perpendicular by sector.
That one down if you join all the centroid.

Okay joint all the centroid, then magically you will get a very nice polygon.
Cool one. It's a mathematical duel. Duel duel of the other from one.
You can get the other because if you're using the polygon that weren't a polygon, find it centroid and then find the next centroid and join them.
You get the triangulation to go from one to the other. Okay. So pretty neat.
That's how all this is born. So just want to tell you. All right, so then when K is equal to one, this is special case.
I going to play region like all this right again any point like in this case this is such that any point inside this is
close to this point as opposed to any other point that is the reason that's how it shouldn't born in the first place.
You know, think of lines in a forest. It's a big forest.
And each little centroid each are these like a lion. Then the question is what should be the lines?
Territory work on the line, Roman Hunt Because when you cross too far, certainly gangs.
Gangs. Even in L.A. Even though it's not a learner.
And if they if the police actually detect actually. Right. The plot like where the gang members are, that is actually what it is.
That is the border between one gang and another gang. That is literally the border.
In other words, this is my area. I sell drugs to people.
Crossover rival gang. If they come and sell drugs here, somebody's going to come and kill them.
Literally. Okay. Don't come in my territory. Merely a line between them.
Right. You're over there. I'm more here invisible.
And see the line. Don't cross it. So you don't. Pairwise two at a time.
Okay. That's that diagram. It's a great data structure.
Tell it over and over again. So then for classification, you already know that each polygon means one type of label, meaning one document type.
Like physics, chemistry, you know, and then physical chemistry.
Then the question is on a new document containing exactly where to call it or when a new query comes in,
you know exactly what kind of documents to serve.
Basically the same thing similarity search the similarities based on what polygon aligned in vector similarity search.
Why don't I polygons. Yeah. Yeah.
Like this for one year. Never. We sent this document to the class of its closest neighbor, which basically means it centroid.
Okay. So anytime comes in, that's a query document, you know, Fine.
Like in this case, it is in that particular in a polygon. I don't know what this boundaries, but that will basically be in this polygon.
It'll be like whatever type this is. Okay. So. We have cell phone towers.
Exactly. You know, so all the cell phone towers right there are centralized for each tower, this overnight polygon.
So if you're inside, you are in a polygon, a drawing, and making a cell phone call. That tower handles your call.
So when you cross the boundary and gone alone, you're just not going to be sure. You say you're driving your car, you're going this way.
Okay. As long as you're still near the cell phone tower.
The call is being handled by this tower. When you cross and go here, your call is handed off to the next door.
The cell phone towers naturally form like overnight, sort of Starbucks.
All the Starbucks in L.A. would wake up and watch Starbucks. Should I go to my closest Starbucks?
You know, church, parish, parishioners. I told you, it's called a parish.
A parish means all the people that live in a certain neighborhood out on the church.
All the examples agree that already looking. All right.
Fun. What else can I tell you? She would do a break or do a break Maybe in, like, 6 minutes or get on 40.
There's not much more else to say. Okay, I can use numbers again.
No speeches are actually necessary. You know, it's like a lazy learner. No training, necessary, nothing.
There's simply no point in your documented to tell you what class it is.
And. Yeah, so no need to train anything at all. And then the classes.
Yes. Are this one. You know, I basically if the points change all that that means is if any one of these points changes.
Okay, so that's more documents that are of certain type, it'll push the boundary.
I'll actually show that I'll take a little break. Okay. Otherwise, depending on what the shape of one polygon is in, like a

zero sum game because,
you know, and then there's only so much down to the whole thing is one way square.
So if you if one becomes bigger so much like the other one gets small, so to speak.
Okay. Okay. I think that's almost it.
Yeah, right. So then in ten years now we can actually compare cancerous neighbors versus actual classic Rock Hill.
So that the reason why that last bullet point exists in actual Rock Hill is all about spherical boundaries.
Okay, You still have centroid document centroid, but you pretend that there's a spherical boundary like a bubble, whereas in this it is all linear boundaries in a polygon boundary.
So then it's actually more accurate that Canada's neighbors then Rockville.
Okay. So then before we take a little break, we can have a little fun.
I'm going to say JavaScript tonight. JavaScript. Actually, I'm going to say a paper that just if this comes up.
Paper are just, uh, examples.
Oh. Look at that.
Should I do what, Alan? Look at that.
So, I mean, so they're all polygons and interactively driving the thing around.
And then it pushes the others to the sportsman by saying when the documents, the clustering undergoes sudden change,
it'll locally affect the neighboring centroid as well, meaning the classification would change just a little bit.
So only. Right. And by the way, the code for all this is like so simple.
There's just a library called Paper Digest. I want to show you more.
I want to show you JavaScript D, 3d3 data driven documents our nine.
D3. Why don't I this? Well, here's Delanie, by the way. This also D3, but you actually want to play with it.
Oh, where would that be? Let me go to D3 Where?
Annoying observable HQ.
Just like your Jupiter notebooks, you can actually have JavaScript notebooks also that's hosted by a site called Observable HQ.
However, 93. Oh, cool. Let's try this one.
I have no idea what all this is, by the way. Never run it. Uh oh.
All of us. This one shows your boat. Seen the dark lines in I think lines they show you a lot of polygons in ten great lines.
They show you. Delaney the boat being updated while they move this thing around.
So, all right, that's basically what it is. So it's always convex polygons where polygons can never be concave.
Concave means a cut to never be a Catholic bishop. Cut. Always look, expand out like a bubble.
Okay? Now, look, this is an odd for this is so small.
In fact, the solid is, believe it or not. Right. Because there's a function called learning that they learned at learn.
I mean, I think it's like random. Every time you run.
It might be a little bit different to watch here. It's going to change. Okay.
And then maybe one last one JavaScript or nine interactive, maybe you can click on each one of them.
Interactive. Interactive Construction.
Yeah. It was in.
Oh. Oh, that's neat. I'm clicking and it's adding some cool points here, right?
I quite actually. What's happening? Huh? Wow. Okay.
Look at that. It's called a sweep plane algorithm. There are many algorithms for annoying.
I showed you one with the Dylan that this was not. It's not Dylan. It sweeps Dylan and brush sweep.
It's called the Steve Fortune Fortune algorithm.
Fortune sweep line algorithm. If you want to know so many ways to do it.
Fortune. Flournoy SWE planner with them.
Yeah. So we're planning. That's what's happening here.
Oh, it's already done. Right. Okay. You can put in, like, you know, I think we should stop here.
Why don't we take another break? So in 39, we break till 745.
Then we can do the rest. Okay. So please be back at 745.
Then we can talk about the last part. Recommendation systems.
That's an easy one, by the way. You know, not that this is hard, but then it's an easier two types of recommendation systems.

All right. I'm going to just keep doing more and more of these.
Step sequencer and.
Right. And I wonder what not to do regarding the last.
I was in my car.
I recall you ask that it's always about the mic.
Drop the keep talking and listening.
In mostly second hand. Where things are right now are really happening.
You think we. It's fun.
It's colorful, fun and useful to maybe take part of every day a little bit.
Exactly. Right. Exactly. For some reason, it's related.
So happy to have you. You talk to me now, write your name down and I tell you the next day,
I basically ask you to send an email asking for an essay and then just go through that and execute it during the break.
During the week. Exactly. Well, okay, no problem.
I. I know.
I know. It's just easier and easier for you to go on.
How many press on? 30. 270.
289. Wow. Cool.
Mm hmm. Mm hmm.
Hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm hmm.
Hmm hmm. All right.
Oh, this is so sweet.
Thank you. I appreciate it. I lost track of the time. I thought there was one more minute left.
Okay, so let's have the last bit of fun for today, which is going to be okay.
This is cool, right? And then this turned out to be a very easy idea.
Karaoke, on the one hand, to make your query more relevant based on user feedback and start talking about classification, which can be okay.
Nearest neighbors or maybe one nearest neighbor. That's pretty much what all of this so easy, right?
And then those easy, but goes back to embedding embedding.
All right, then. Last stuff.
I still have almost 40 minutes below 30 minutes. Okay.
It's pretty neat. All right, so then what we should do is this.
And because I have a little extra time I wanted to treat or this actually is great.
I told you this last time, Professor Emily of Harris paper, where it talks about all the horrible things in the world that people can do.
A genocide in Ukraine, where Zelenskyy was made into a video where he said, let's surrender to Russia and obviously not right.
Hamas, Israel war. All kinds of crazy things that are being made editorially of people saying screw Israel.
Long live Israel. Make people say exactly the opposite of what they said.
Right. It's horrible. Okay. So but that's not what I'm going to tell you now.
I'm going to tell you this RSS is so interesting.
It's a way of doing, again, an information retrieval account. So they still exist, by the way.
I'm going to show you the US military. Right. If I start my own this is a B 21 bomber and I have this right for talks about larceny things.
But at the very end, this thing called RSS, right? It's called an RSS feed.
It's basically some kind of an symbol document. Okay. She looks like that.
Exactly. So that's how we used to show information to people. It's called an RSS feed.
You are special apps called RSS readers. And then your diet is that you take the link that I'm going to show you and paste it in the reader in a format all this and show that information to you.
Then people can every day push information and the reader will automatically trade you.
It's called feed reading. You're subscribed to a bunch of feeds. You subscribe to a photography feed recipe feed.
In this case, military feed. The feed or medical update is a cool idea.
Okay, so let me show this to you. I'm going to copy that same link that I show to your copy link address.

But then how do you make it look? Be real. You would go to, for example, go here and paste that keyword, paste a link right there.
View RSS. You really, Really.
Look at that. Whoa. All this was in the link. All this wasn't the index.html file.
It's just a nice way to do information, search and retrieve. You should be excited, but it still works better.
It's very low bandwidth. Easy, beautiful RSS. Likewise, you can also have pasted this onto a different this actually very cool, by the way.
It randomly goes some place in the world and then grabs the RSS feed and then shows it to you, you know, in Google search.
Right. It says, I feel lucky, selected. Okay. It's neat. Just look at this.
So it's an RSS viewer app. As you can see. You don't even know what you're going to get.
But, you know, that's the fun of it. Your idea just thinks that the world even exists.
But I want to go to this one. And paste the same link that we had to over here.
Johnny Depp. Somebody called Johnny Depp. He made that site meaning he wrote the code for that.
Get RSS feed and then it's going to show you the exact same information.
Right. Because that's what's index.html look a link to a page, a file, all the text.
You know, it's a neat idea idea called RSS. Someday you can use it to share information within your company.
And these days you can have a little apps that run on your phone. But RSS is lightweight.
It's a cool idea. Okay? It's now been basically forgotten by a lot of people, but it's still something quite relevant.
Whoa. By the way, about our Facebook.
Crazy, right? Whoa. So many seeds.
I have no idea what any of these are. Let's go to our sister Canada, as some have broken it out, clearly.
But let's go to the top ones, maybe the ABC News. Right. US headlines.
So what do they have to say? Oh, okay. So we should grab that and go back to one of the viewers.
Right. And paste it here. Let's do the rest of the first viewer and we can move on to our recommendation system.
Okay. Latest news, I'm sure.
Whoa. Hey, President. She was in San Francisco yesterday from Crescent City.
Yeah, I mean, all kinds of, you know, news about the world.
Sony tried it and consume news in a very unexpected way through all those RSS feeds that had no idea about.
Great for another time, but you can go click on little. Can you do one last on CNN, right?
Obviously it is CNN. So you and CNN to this day that take all the news about Israel, Gaza, for instance,
I'm pretty sure we'll actually make an RSS feed because there are some people in the world that still depend on this very low cell phone connection,
low bandwidth.
This all look very easy to get across again, whereas your app with videos and like what what right is like pretty hard like heavy bandwidth.
So we in some countries in a way Internet is not easily available.
Wow. So again, to give you the text texting only when you click on it and show you something a little bit more.
Now, we would actually take to the streets in the comic. All right. So that's a pretty easy way to serve and search for information.
Okay. Recommendation system, another fantastic piece of information retrieval technology.
I'm glad we're just chomping through all the rock. Looks so easy. About 20 slides.
We'll see how far we get. Yes.
So if I know about something like I buy a book about machine learning, Amazon tells me people that bought this book also bought.
You know. So what else about we should actually go do it.
Okay. If you go to Amazon dot com you know. When you search for anything, for example, search for JavaScript, the good parts.
Okay. There's a book called JavaScript The Good Parts Java Script, The Good Parts.
But the point is the recommendation engine obviously is going to find that book right as the book.
But there's also all the critical. More results, More results, and then even more results at the bottom somewhere.
Yeah. First of all, this one related search is right.
But now say actually that's a formal recommendation. Say I actually buy one of the books.

Suppose I by an actual book called The Ultimate Hands-On in a Machine Learning with Plato's book.
But now we see like the see this recommendation frequently bought together.
So I did not know that the other two bookshelves are pretty cool. So hard to know what does the clustering it cluster the three books together?
Where did the clustering come from? People's behavior. People frequently buy those together, people that know what what those books are about.
I did not know about these two other books now and say, Hey, my list will add to my cadre.
Also this. And I feel like deep learning. You might like other deep learning.
You end up people do not buy them. You might like more books. Does entertaining this coming from clustering.
Look at that so-called learn more and more data or even this one, you know similar items, literally similar items.
Cluster, cluster. Wow. Okay, so that's what I'm going to show you.
So where does clustering come from? Here to have. Okay, There's two big algorithms for doing recommendation.
One class of algorithms. I'm going to point it this way. Does not actually care about what is being recommended.
It does not care what's in the book, does not care what's in the pizza, does not care what's in the movie,
does not care what's in the song, does not care what's in the medical supplement text.
It's all so called from the outside. That is the one form of a recommendation algorithm.
It is called collaborative Filtering CFA.
We call it CFA. On this side, you actually look inside the food and say, this one is heavy in carbohydrates,
this one is sugar based, this one is high in protein or from music.
You would look at somebody talking on I based on Dr. Grass, I know it's talking a little distracting in the music.
You will actually look in the music and say, this is a blues music, This is electric blues.
This has got like a fast drum beat. This one is a jazz song, this a composition and minor.
You have to actually know inside the music or for a book, you have to know the contents of the book.
This is a murder mystery novel where it's like, you know, not too much suspension, but whatever, right on and on or in the machine learning.
It will tell you this a book about deep learning. Once you know the actual content, it is called content based filtering.
So collaborative filtering, you see if content based filtering is CBS, Netflix, for instance.
Tock They would combine both of them, then that's even better than just doing one of them.
So real world accumulation systems combine them.
Okay, That's basically all right here. Okay. Collaborative versus content based.
Again, the big difference is this all based on people and items together.
The only two humans consume things. We read books, we listen to music, we watch Netflix shows.
All right. So that humans items means, again, books, food order together.
Right? But it's people that do that. You can have a matrix on the one hand.
Person 1%, 2%, 3% for Netflix has 140 million subscribers on this other axis.
Each TV show, each movie that you can watch on Netflix, that's a nice matrix.
Okay? In that matrix you can follow at once. And zero one means city.
Eric Clapton. Yes, if I heard that song.
So one have not heard that song so far in my life. Zero.
So imagine the Matrix. But again, hundreds of millions Frozen columns, right?
It's amazing. But that Matrix exists actually. Netflix, that's what they do.
That's actually what they mine. Then when a new person or even an existing person has their their pattern looks similar to so many other people.
I said to draw one of the pictures we will draw here.
Supposing you have user a this one user account user and and these are all the different songs you can either listen to or not listen to.
User has to listen to them all. So all 111.
Actually no, it usually has user support here.
User B that for some reason has heard this, listen to these songs and not listen to them.
Do not listen. Yes, listen. Those two are so similar in terms of cosine similarity, these two vectors are almost identical.
Then you say, you know, if you ask, would this person be like that particular music again,
the music titles going, okay, so should I make this part of the recommendation for B or not?
The answer is almost unanimous. Lord Yes, because, you know, it's a lot like the other person and not only the other

person.
So there are many, many others that have a pretty similar profile then that is one you could almost be sure that this person would also like that,
because that's that's a behavior, this 11100 that and you can even refine this year and more if you go and vote and rate things,
then you can even take that into account. Like all these groups of people rated all these movies pretty highly.
This not that they didn't watch them, they watched them but hated them. And this person also hated those same movies.
So then they're even more similar to other people.
So it's all about users and production, like a matrix that is all your network can learn that pattern.
By the way, this just only for these three for you is this right? So many others just will have different ones and zeros.
So in the end it's a big matrix of ones and zeros. And your network can learn the pattern, the ones and zeros.
Then start recommending to somebody in that ask a question, should would they like this movie or not?
The answer is yes. It'll show up literally in this list. In this list.
Over here in the list. So in all these products related, Right.
In other words, on the one hand, it's based on obviously, you know, these work together.
If I sign in, it'll also take my preferences into account that will actually intersect the of.
Okay. And that's where something like this comes in. Okay. So I'm going to tell you all of that now.
Okay. So online systems have lots and lots of good recommendation is a very big deal on your phone.
You only have so much scroll space. This cannot be infinitely scrolling.
Can look they only get one chance to basically wow you.
If they show you what's relevant, you will hopefully they'll make more relevant things will click away.
And so what? Where are they showing me this? That is why you have to pick it.
Well, so Ticktalk, I told you, I think one time Tick tock has a recommendation and you know, item called monolith.
I mean it's a magical little thing is go okay some monolith algorithm that realtime recommendation system I think one time I
told you this I remember the use a very cool thing called coalition list embedding basically using two different hash functions.
Normally people use only one hash function that goes into one hash table.
There's two different hash tables.
And if there's a hash coalition in one of them, that means if some of their in a value also goes in exactly the same location,
meaning the hash function sent two different values and me to the same hash location.
Then what'll happen is the new arrival. Colby will kick out. New call that is coming in will actually kick me out.
He's like a bird. Cuckoo bird will lay eggs on some other bird's nest.
Are too lazy to build own nest by John you know kick it out then that for a says omelet for business.
Wow, where the heck do I go? I'll go and find a space in some of the other data structure.
The other table has to rule. Sadly, if somebody else is also living there, they'll get kicked out.
It's all taken out. Happens like a few times. And then finally, if you cannot resolve it, then the two of you can live together.
But by kicking that out, they can actually make the whole thing have a higher throughput and also a better recommendation system.
This is tick tock of all people, right? So cool. I mean, look at them.
They're all by bytedance, every single one of them except for that all the way by tens.
So they beautiful. In other words, it's a pretty big deal.
The US government. Okay, why the heck are anything like US children? Because algorithm is so powerful.
Okay, recommendation algorithm, which is a big deal. Really. So we should care.
Okay. Yeah. So you get it? Okay. You can discover new products, you know, also recommendations that are pretty similar to each other.
They can do what's called a filter bubble.
If you go and read too many far right news channels, okay, then certainly the system will recommend you even more of far right radical news channels.
It will go further down in the rabbit hole and pretty soon you'll only see people that are exactly like you.
You lost touch with humanity. That's a pretty real problem.

Okay, the whole recommendation idea. So always keep that in mind as well.
In any case, all of this bias, again,
all of this is still called collaborative filtering because it is all about other people's behavior, not about contents.
I'm not going to tell you that yet. Okay. So two types again, one is content based filtering, one that says classic content based filtering.
Yeah. Okay. It means you got to look inside the actual song and can even tell you how collaboration is about user behavior.
Okay, Lazy and neural hybrid. That slide is pretty simple.
All right, let's start from Pandora.
Before Spotify, they used to be three different Internet radio stations that all did exactly the same thing.
They'll build a custom playlist for you. Somebody is talking. Please don't talk.
Please, please. One last time. There's only 20 more minutes.
Okay. All right. So now all three of them build custom playlists for you.
So you listen to a whole bunch of songs, right?
So of millions of other people, they'll compare them all with each other and then make you all be similar, so to speak.
In other words, find somebody more like you and then recommend to you something.
They listen, but you're not listen yet and make you a listener.
Yeah, sure. Like this right station.
I recommended songs by observing what bands and individual tracks the user and then comparing that particular user's behavior with so many others.
Similarity search. It does not get better than similar search. Yeah.
So then. Yeah. The whole point is there are so many others like you and never listen to certain songs that I have not listened to yet.
The curated line it up. I think you would like this. You know, you listen to that song, also listen to people that but that product also bought this.
People that bought a laptop also bought a laptop case. I did not even know that I could buy a laptop case.
People that bought a laptop also bought an extra battery holder, but a laptop bought a pen for 30 extra dollars.
People that bought a laptop but glass screen cleaner on and on, on things.
I had no idea that existed. Okay. But they're going to tell me wonderful Pandora.
Okay. This collaborative filtering just simply based on other people's behavior.
This is the one that is content based. There's amazing there's something called a music DNA project.
Imagine for any song, whether it's carpenters or, you know, Beethoven's Fifth Symphony, 200 musical attributes, you have to have a musician.
You want to know there are 200 different things I can measure about one song, any song, and that's a column of 200, 200 column sector.
Okay. And each song will have either yes or no for each of the 200 attributes that is called a music DNA project.
And then they can use that to classify all the existing songs and do blues, you know, guitar.
I told you that just then. When a new song comes in, they'll know exactly what to classify it as.
Likewise, if a user has listened to a whole bunch of things that look like blues,
the look in the song DNA database and find other blues songs and recommend it to do not based on whether other people listen to the blues or not,
but based on actual music. You listen to the look inside your music and do I recommend it?
Do a similar research with the music DNA databases so that you can look at this music DNA project,
music DNA project, because the music itself can have DNA Music Genome project.
Yeah. He's got so many names for 50 genes.
Look at that. Any song can have 450 genes.
And then, of course, rock and pop in order to replicate only 152 basically classify every song ever,
and then all these stations would actually use them. It is pretty neat.
And they got a patent. I think Pandora did look at it.
Yeah. Music Genome Project like this.
Fourth of all, you have to bring together every track, any song up to 450 attributes, and then you can use that to filter.
That's called a content based filtering item similarity as opposed to user similarity.
Okay, so then tell you more or. This is a simple example can.

Maybe this restaurant in Palo Alto. I'm not sure. I know you'll find us in Palo Alto.
I looked in the next street conference in York in Palo Alto.
Okay, so imagine this someplace. And people write salads like sit in restaurants.
Alice likes l l Alice does not like Street's cafe.
So got a bunch of people, got a bunch of restaurants again, remember, people can go in the Y axis, restaurants going to the x axis.
And this is my matrix of zeros and ones, you know, check table.
I can make that in the matrix it out. Okay. Then from the Matrix.
What Alice like some brand new restaurant are not and other people are already gone from that collective behavior.
They'll either recommend Alice the new restaurant or not. Wow.
Amazing. So again, I recommend to you the popular restaurants, positive minus negative, you know, So again,
you can ignore your completely and just simply take all the other restaurants you're not
eating yet and look for the good ones in a positive minus negative to bring them up.
But then they can also take your preference into account. That is when the matrix comes in.
In and otherwise, there's two ways to recommend.
One this ignore you completely and just simply recommend whatever to help you embarrassed this metal point that says,
you know, take your preference into account and then also then use that.
Great. That is the notion of like minded people.
So then just to make an assumption, right, there's some reason why I keep on listening to idioms songs.
Okay, I love William is not random, and that's definitely a good assumption.
Like we don't do it to throw the recommendation engines on, like we don't care about that.
But instead there's a reason why we like certain things. Yeah.
And then likewise, you know, if we look in this table here,
almost anybody that liked a certain Italian restaurant might have liked other Italian restaurants as well.
Okay, then the idea would be then if you like Italian restaurants and most likely you'll also like the new Italian restaurant in town.
That is the whole idea. Go look at all that. All right, so Senso, in this case, there's a matrix.
Look at the matrix can be highly incomplete. And that is the whole point.
If there's a pattern in here, and that is, would S-T like mango?
Yes or no? It's an open question. And your network and answering so all these blanks can be filled.
This is a very tiny toy dataset. Again, Netflix has almost 200 million subscribers, tens of millions of TV shows and movies.
This matrix is giant. I don't know if you know, Netflix is data science salary.
More than half a million a year. Okay.
Because that's what they pay you for, to keep this going really well and then have Belichick talking aggressively.
So Netflix. Data science data not data center boring data scientist salary.
Whoa. About half million people here.
It's crazy, right? I mean, that that's insane, because the recommendation is everything.
That's a whole business. Okay? Oh, it's a big deal.
Okay. And Netflix has a blog in the blog.
They most certainly talk about the recommendation systems to give away some of their secrets, but not too much.
Our list of almost 50 different blogs, engineering blogs.
I'm going to put it up for you next to us over the weekend. Okay, See that?
You can actually read how they're doing it. There's a personalization.
The rest, you go to sign that sign and I have a medium account by $5 a month just paid and get the pro account.
It's like, so valuable. All right.
That is called the utility matrix account is simply the ones and zeros that simply tell you the user's past behavior, existing behavior.
Then that would be what, Fred? Like California, yes or no? The blanks are used for the recommendation at all.
There's no need to already recommend this, right, Alice? Already, like when you get the idea.
Okay, so for any one person, take their existing preferences and fill in the blanks as intelligently as possible with the new preferences.

Guess intelligently. Guess it's an easy problem. Right? All right.

So then you again, use those items again expert versus y and make a matrix out of it.

And then yeah, you can again have a not just a yes or no, but yes can be in a scale of 1 to 5.

You know that no means easier. Okay.

So I can turn this into not just a binary label of zero or one, it can be a multi multilevel classifier on a scale of 0 to 5.

Then these numbers also would then be calculated on a scale of 0 to 5.

That can be decimal because no, it's averaging them. And then suppose Palace XL restaurant recommendation is 2.2.

You might have a threshold that says anything below three.

I'm not going to recommend and I can turn that into some kind of threshold as opposed to binary.

True or false doesn't make any sense, right? Yeah. So it does not have to be zero or one can be on a scale of 1 to 5 like Yelp review, you know that.

Okay, you can take your I point of view and also other people will produce and train to the recommendation.

So yeah, exactly like this 1 to 5. The matrix is sparse. Okay.

I mean, that's the whole point. This method is for we have nothing to recommend anybody.

Okay. But the whole point is there are more things in the world than users.

If I go to CVS, I've not bought every single gosh darn thing in CVS.

Okay. All right. Target already. Outrageous.

So that is why I have this golden opportunity here that the store has an opportunity to recommend all kinds of things to me that had no idea existed.

I go by the same shampoo over and over again, but it tell me, you know, this conditioner, many, many people buy this conditioner.

Why aren't you buying it? Serious prints like mile long coupons. Okay.

If you look at the coupons, that's exactly what the print If you look at what you bought, it goes hand in hand.

You bought something, but you did not buy something else at other people.

But it's in the coupon like coupon because attempt you, if they tell you buy it for the full price, you would have.

But they say in air conditioner $3 off and you try to negotiate this pretty good.

So next time guess what you become the person that's buying the two together. Okay.

And you don't mind that it's not cheap anymore. They're basically hooked you. And that's why the initial coupon, I mean, it's all very clever.

Okay. All right. Yes. Then the unknown rating means, you know, exclusive information.

That's what you're trying to get through machine learning. Likewise, you can also do negative voting if somebody actually does look up or down.

Right. And they can do that as well.

And then once again, then the recommendation might come in like a decimal value and then you can say anything more than zero.

I'm going to say negative. You could read through all this a little boring.

In the they're all simply vectors similarity. Okay. The whole thing can be a vector and somebody else can be a vector and that non-zero is okay.

Then you can actually guess what the zero value is going to be. I'm going to similarly switch.

Yeah. Again, like I said, this can be like fractions, right? Because they can be like rating scale.

Yeah, that's the whole idea. Make inferences from user behavior.

So you can say there's a reason why Alice loves our talks about like science fiction kind of music and see,
you know, I don't know what the reason is, but some reason why those numbers exist.

And then yeah, that means she likes this movie, like, pretty strongly.

Okay. Like all of that. Okay. I'm going to skip some of these details and then look at this stuff.

It's all a game to Sam. Victor Hoyt took the whole content and the words.

One role in in the one on Lazarus is like some kind of a victor.

And I can do similarities, one with missing data and still do as Mary research.

Yeah. Important words in the document. So now I sort of a document recommendation system.

So what we're saying is I already like a certain document or my query is considered some kind of a, you know, initial search.

What other, like what related documents? I'm trying to search for some keywords.

Okay. So what are you going to recommend me? But now it's not just simply index based keyword search.

It's more like a recommendation system I'm going to search and how we're going to do it.
Still, we use the same two ideas and then you have this vector of weights, every document, selector of weights.
This is why you can take a gain vector database like pinecone, and then take the entire document,
your whole PDF file full of sentences, and turn them all into some kind of a collection of vectors.
Okay. And then that can be used to lecture people. Yeah.
So these are all things I've already told you. Okay. So then again, once again, this is how you make the weighting between I and J.
So all of these were gone through before, but this is the main idea.
So the content itself is some kind of an array of these kinds of weights.
Okay. Satisfied of number of documents. How many times? Yeah.
So all of these are classic DFT of definition, really, even in this kind of frequency.
Right. Again, I told you this in the past where a frequency is for any given keyword.
How many times does a keyword occur in a certain document? Other words, there are so many different keywords and so many different documents.
What is the vacancy of a word in a document? Conversely, a document where basically a matrix I can go in and fill in.
Then I can recommend you start to recommend people, because when when your query comes in,
it'll also be some kind of a vector exactly like this then that can dissimilarity search.
So all about similarity over and over and over again. Okay.
This, this matrix can be boolean meaning like don't like or it can be on a scale of 1 to 5.
I'm going to start telling you that. So once again, this actually what I try to do over there.
I can show all the oldest movies and actors, actually.
Okay. This one is also very interesting. You can even search based on, you know, different movies and different actresses.
What are people like? In other words, actors is a weight.
I mean, I'll let you go figure all this out, but not all people acting all Moyes Right.
I can go to IMDB and make a matrix exactly like this. Take the top ten actors.
Okay. The top hundred Moyes. And say what? Actor, Actress, Actor, Dancer.
Movie. Yes or no? Then I can use that in a way to actually recommend certain movies to certain people.
Because say certain people, like certain actors. Actors is a lot.
Then I can start with that starting data and going this matrix and go back to an actress that they like what Moyes did back then.
And then of those Moyes, what are the use of watch Are they not watch comes in so can jump from one to another, meaning users to actor actresses to movies like that.
If I skip all of this.
You know, there's a reason that when this is simply detail in I read it if you want, but I just want to get to the really cool part.
That is a cool part. So and then the cool part is still all about similarity, you know?
So a user profile X is all about the movies that they watch.
But there's also the actor profiles, meaning the movies that to watch how certain actors and actresses,
then they can use that extra information of not just the movies that somebody watched,
but also because movies happen to have certain actors and actresses then go down the MTV movie database and searching for other
movies that have the exact same actors and actresses that this person has not watched yet and then recommend that to them.
And that might be a higher a higher probability that they would like the new movie I'm recommending.
Why? Because I happen to like certain directors. Nicole Kidman.
Whatever. Al Pacino. So then.
But, you know, my my point really is we keep going back to this idea again and again and again.
It reappears in so many ways. You know, one way you can summarize this course is this okay.
And this goes back to 1950s user behavior to teach people to watch movies.
Okay. And 50 people in the store. And what things in 50 people read books.
People search for documents in the library.
The coolest thing that people found the six community discovered information or to a community discovered is the notion

of tracing everything.
We discovered somehow that an X and Y and Z and W and thousand dimensions are preferences are clustered.
The way we use certain words to describe certain things are clustered, products are clustered.
You know, everything is clustered, cluster clustered.
As soon as clusters happen that that that we can draw boundaries on the cluster and then we can do classification after that like a dissimilarity.
Such a new dot appeared. So what other dots are nearby?
That is the core idea. That is where you see it again and again and again.
It's 816, so I'm going to do four more minutes. Okay, Hold on. You can go ahead.
What can I show you? All right. So the notion of content based approach, you know.
Okay. So we can compare collaborative versus content based and content based.
We actually don't care about other users. If I like a certain kind of music, I like fast EDM.
I like trance music, for example, right? The nice trends get thing going then.
Then the system can look through all the other songs and get similar music.
For me, it's not based on other users at all. And then that way, anybody that has the most unique taste, there's going to be some other song,
some other book, some other movie almost in the same genre, and the system will go and find it.
Why? Because we look deep inside the content itself and that's pretty neat.
And then beyond popular items, this is actually pretty cool. One big problem with any recommendation system is if nobody recommended something.
Samson. Sounds like so many things, so many flash drives, you know, so many like all these things.
Sadly, nobody ever recommended a product. That product would not be bought by anybody else.
It's a catch 22. That manufacturer, that person that sells a scooter, because, you know, that's called the first it's called a cold start problem.
Somebody has to start recommending and I will try it. And I do recommend up or down, be viral and just die.
But if nobody touches it, the recommendation is zero. It never refactored anything.
If you use collaborative filtering, but with content based filtering, thankfully even the most unpopular item, somebody like something like that.
And so then the system can recommend that for you. Okay. That's a pretty big advantage of content based filtering.
Great. Yeah, exactly. And then also, this one is pretty cool.
You can even actually ask, why did you recommend me this? Well, because I is 120 bpm.
You're like 120 BP music a lot. There's almost like, explanatory.
I usually recommend something. And so why did you do that? Well, just look at more nuance.
But that tells you nothing. But now you can ask it, you know, Why did you recommend me this?
You can actually tell you, okay, it's called Excel. I you know, maybe I should stop you.
Me? Meaning? I'll let you go. But I'm going to tell you this X A.I. stands for Explainable A.I.
So this might be an example of where you want to ask the AI.
Why did you make this decision right? Why did you, you know, classify it like this?
Why did you get this person then? It'll actually tell you, hopefully in human terms.
So in that sense, content filtering can actually give you explanation.
So why don't we stop at this? Still only a few more. Okay.
To summarize, I'm going to give you the homework for the weekend part and part B for like for you and for me.
And then there's no class. Next Thursday, we meet one more time and wrap everything up.
Okay.

Lecture - 4
Sorry I'm late. Okay. Not intentional. Hey, cool.
So I have something hopefully good to tell you. Which is, uh, your lectures from last week.
When I was so helpfully away or actually on duty.
Well, see, if you log on to a part of the portal, take a look.
It's had two videos from last week from last term rather so last term, fall 23.
Lecture number 12. Lecture number 13 Q&A clustering classification recommendation systems.
Wow, I probably almost worth the same lecture I don't think I did you know.
So it's all in there. And without missing a beat my song you can just jump right in and catch up.
Okay I'll only tell you a few words about those just to, I guess, tie everything together and then we'll do a little bit of new stuff today.
But on purpose I call it assorted assorted Topics part one, because then next week I can do assorted topics.
Part two. I want to tell you 2024 things.
What happened last week? Actually, believe it or not, but not today.
I'll tell you that next class I sorted, because I can then name a whole bunch of them and then point you to a whole bunch of resources.
Because this topic. Right, information retrieval has changed.
You know so much. Uh, since ChatGPT is simply dropped from the sky, which is November 2022.
So it's not even two years old. And so everything is being still revised.
We still don't have best practices, you know, in programing languages.
Now we have best practices. So we know the Llvm, our compiler, for example, the the process, the technique.
We also know about like object orientation functional programing.
But then when we transition from a low level programing like assembly to a high level like Algol,
you know, C, Fortran, snowball, all those languages, we had no idea what would even work.
Okay. Tried like a whole bunch of things. Likewise with the world.
Even though are only two years old, there's already so much change in how people actually approach things.
So already some of the things I'm going to tell you, they are not obsolete,
but they're a little bit old fashioned, like given the terminology I used last year.
Wow. You know, so I show you all that. But first things first, I said to you Q and a clustering classification recommendation systems.
So the videos for all those are out there from last week. You can go look at it.
So we're not behind okay. But please watch it though. And then for the exam.
So whatever I said in the video I pretend I said them here. It's the same thing okay just watch man video.
That's what then people do anyway to watch me in radio. So last week we're on done.
You had no idea how sick I was. I hardly cancel class. Okay. Um.
Throughout the pandemic, throughout 2020, and all through Covid cycles, multiple waves of Covid that swept through the world.
Yours truly never got sick. Okay, I didn't even get a cold.
I never got a flu or nothing. But I guess you can only dodge these things for so long and finally does catch up with you.
So for me, emails last week okay, yeah, I couldn't even basically talk actually was like so bad.
And, uh, I hope not to use these things, but I've got a couple of secrets.
I'm not. Free cola.
Except that goes Ricola. Ricola!
So not to cough too much. Okay, so got my hook.
Come prepared to. Got some cool things, as usual.
You know, some fun things to interest you. For example, in the order of increasing success.
In the order of increasing effectiveness. Those are the best ways to learn ML slash data science.
Okay, so pay attention to what's on the bottom list. The best way if it makes you smile, if you feel happy when you're doing it, you know.
So it doesn't beat that right. Yeah.
So this one you know see that know you know I do some machine learning to you know so goes and imports the Python library.
Yeah. So TensorFlow you know maybe on this meme you know what's made TensorFlow is basically the top of the pile

okay.
But it's no longer because you actually have things like Keras and also PyTorch.
So correct. And speaking of low level high level at one point TensorFlow.
What's the high level in TensorFlow? A tensor is just simply a multi-dimensional list.
And so you write only one kind of function ever.
And the function always takes a bunch of tensors going in, which is a bunch of multi-dimensional lists and always outputs exactly one tensor.
Show you, and then you can program self-driving cars with an so initial version of Google's Waymo car.
It runs in hardware. It's called TPU tensor processing unit. Why tensor?
Because in Python you will list a list and then you have a list of a list.
Right. And then you can have lists of those.
Meaning you can have a list of a list of a list.
That's a three dimensional list. You know, one dimensional list.
Two dimensional list. You can keep going. That's all tensor flow us.
So then Google ends up making a whole Python library where every function by function I mean a neural network, uh, multiplying an incoming feature with it, doing a dot product. It's a TensorFlow call.
Tensors going these kinds of tensors. The going could come from the real world traffic.
And then the output would then be some kind of dot product.
But then that can be combined to a different TensorFlow function which also takes tensors and tensors out.
This is called training data flow. It makes it easy to write code because everything is only tensors and tensors out.
And they made a graph of it. Made a graph. It's a graph.
Check this out. Okay. Okay.
See? Imagine is doing something cool, right? One input.
And then that passes the output to this function that has no inputs passes the output of that function.
And then maybe here's another function. This also passes the output to that function.
No input meaning real world input and output is called a graph.
TensorFlow. Everything is a graph. So then you run the graph.
You want something at the very end, but it's not a Python program. You don't put on the whole thing in a big.py file and run it.
The problem with that is it will start from the top line in your.py file and basically go through piece by piece function called side,
but it cannot run something in the middle. In a graph. Here's what you can do.
What TensorFlow does is suppose you run this whole thing one time and something will happen.
The traffic light is green, the car is going. But then say something different happens here.
A pedestrian suddenly runs in the middle of the road.
The crazy cool thing is this in graph programing, because this changed, something changed here, right?
This goes and says, hey, you need to change that will then compute an output, and then it will tell this function how you need to change.
You need to change your car or hit the brake. The best part is this does not have to rerun.
This also does not have to rerun. What if this was a tiny gigabyte file that took like, you know, half minute to load?
You don't have to do all of that again. So in graph programing you can only run the parts that need to be run.
This is called damage. Only the nodes that are damaged need to be recalculated.
So an amazing idea okay so even better imagine something at the last minute changed then only that function is recompute.
Whatever. Recompute all of this part does not have to run. That is the difference between a static script and what is called a graph.
So TensorFlow used to be a big deal, but TensorFlow takes lots and lots and lots of calculation.
A lot of code rather. So look tensor graph. So TensorFlow actually I'm going to say TensorBoard.
Okay so TensorFlow you write in Python code right.
And wired up like that. But you want to visualize it and say hey what does my Python code become.
Then you make this thing called TensorBoard. Look at that.
We directly don't program TensorFlow like this at all, although that would be super cool to do.
Okay. So opportunity for somebody right there.

But that is an after the fact visualization you are script that contains this graph will then be automatically visualized by the system.
So you can know what the connections look like okay. Anyway that's what TensorFlow is from there.
But TensorFlow was deemed too low level because nobody writes all this.
That's the atom optimizer by the way. The standard machine learning loss optimizer.
Adaptive loss right. But it's all been done over and over again.
So why should we deal with it. So Keras Keras front end.
It's a high level front end to TensorFlow backend,
meaning that you write code in Keras and then automatically it will get converted by this module called Tf.keras that backend to good old TensorFlow.
So we don't have to write TensorFlow directly ourselves because it's a pain in the butt.
Likewise PyTorch front end and then TensorFlow back end.
So TensorFlow used to be like that right? Torch has many backends.
One of them is TensorFlow okay. What you used to be.
High level is now actually low level.
People keep on moving to higher and higher levels as a beautiful part, but large language models first came out two years ago.
People made the promise bigger and bigger and bigger, and people even made this thing called prompt engineering or prompt engineering.
You didn't know. But all that is basically gone. Okay, so don't even waste your time trying to be a prompt engineer.
There's no such thing. But instead prompts have become smaller.
You know exactly the opposite way, which is actually a better way, because then we can combine the prompts anywhere we want.
Like a programing language. You hardly write the whole thing as one big prompt, right?
It's not one big function. You write little functions and you combine them.
Like once you write little prompts and then you can chain them. So the whole language chaining that's a better approach.
So things like a line chain okay. It's a better approach.
So like I told you things that are you know that used to be cool like just you and two years ago Intel is changing.
So if this person does this, you know, they probably shouldn't be doing that.
They should be doing import Keras. Keras is awesome, by the way.
In Keras, what might take many hundreds of lines of code in raw TensorFlow?
It might take literally one line. Suppose you want to make a ReLU layer, a max pooling layer, or write one line.
So Keras program for neural network has like eight layers.
You know, Keras program might have nine lines of Python, literally one one layer, one character, one piece of code.
Combined combined command is so beautiful. You should look at all of that.
All right one speaking of clustering. Right. So this might be in the video.
You know that I didn't show you. Um, thankfully what has happened less in the world.
The things we talk about like geography and or chemistry or whatever, right.
They're all like, entirely segmented, the kind of vocabulary you know, we use, right, are all very different from each other.
So when we write words with term scientific terminology, the scientific terminology also can separate like that.
That is the basis of all information retrieval by the way.
So we know that all the books about astronomy a whole libraries, what books of astronomy, if you embed them in some kind of multidimensional space,
they'll all become the green dots, because all of them mention words like, I don't know, telescope wavelength, you know, um.
Black holes. You know, gravitation, you know, space, time warp over and over.
Same words. But then you talk about law books.
Then they would all become the, you know, magenta color on the right hand side without things like, um, I don't know, adjudication jury.
You know, major, you know, amicus, amicus briefs, you know, jury selection.
So use an entirely different terminology. Right. Well, lucky that the world is like that.
Otherwise, no clustering, no classification, nothing is possible. So it turns out that we can cluster things pretty much

based on their words.
So then this is one cluster. Second cluster that was words cluster and clusters thankfully have boundaries.
If you have a law about international space travel and most certainly it's going to be in the middle, right.
There's some kind of border zone between two countries will go back and forth like in a special cases, outliers.
But in general you can properly situate yourself in one of the regions.
So when we talk about document clustering,
that is the video that I and I'm going to have you watch all of the documents of a certain kind will cluster here.
All the other documents of a different kind of like maybe physics, chemistry, physical chemistry.
Physics, uh, biology, biophysics, you know, the the innovation.
Then when a new document comes through, what is the point of all of this?
When a new document comes in, you have no time to manually look at it and go put it in a pile.
Automatically, the system can figure out what topic or topics that document belongs to,
and then add that document to that topic to the cluster already.
So automatic classification okay, that's a great idea. Likewise.
If you upload a document and say, can you show me more and more documents that are pretty similar?
It's a recommendation engine. Then it does the exact same thing. It knows what cluster your document fell in,
and that's like a similarity neighborhood search and then retrieves the closest documents and gives it to you.
So the idea of this kind of classification clustering recommendations, they're all the same idea over and over and over again.
You should never think of them as completely different. They're all the same idea.
It is that we can thankfully geographically separate things in the world.
And once we can do that, we can do magic with it. Cool. So that is the whole K-means little job for you there.
Okay. So K-means, because you're asking for k clusters.
Okay. Yeah. You might even know this algorithm from things like data science where you have the data here.
Okay. The data is all here, but they're not colored in any way. You automatically want the coloring to happen automatically.
When you ask the system, can you please make four different clusters out of this?
Okay. The number four is input value. But then how does it know that this is the boundary?
How hard is it now that is the boundary. That is the boundary. That is the boundary.
It runs through an iteration automatic basically randomization algorithm meaning that you want for clusters right.
And every cluster has what's called a centroid which is the mathematical sigma x divided by the total sigma y divided by total.
So here's the centroid. Here's the centroid B.
Here's the centroid. Here's the centroid. Once we know that we can stop because that is what the user wants right.
But then we don't know that though. So it makes up for random centroids okay.
It might even put one centroid to be over here. Second centroid to be over here.
Trade center centroid here for centroid here. And do this iteration meaning make the centroids move through every single step.
Just move them up. And after a while magically they will all go exactly where they need to be.
And then if one centroid was here second centroid is here.
You join them and you perpendicularly bisect them. So that will be the little border okay.
So everything is then given to you from then on. What if I do.
I wanted attendance just yet, but K-means, I'm not sure if I say K-means like K means we're going to find out.
Okay. Oh, K-means. Okay, so check this out. This actually what?
I mean, I made a little sham. So you can play with centroid one.
Centroid two. Centroid three centroid for decision centroid file one for centroids okay.
And I think I hardcoded all this in K-means. Otherwise okay. Change it here.
But the idea is that these points these circles they don't move.
They're your data points okay. They already are where they need to be. But you're just saying where are those four different centroids.
I'll run it each time okay. Watch it. See that?
So every time I generate random numbers, meaning random number of circles, but each time it will properly find exactly what the set of centroid is.

And that is the migration that I talk about. Fake centroid. Fake centroid.
Fake centroid I guess. Fake centroid.
All fixed and right. All of them move over a whole bunch of iterations and then settle to where they need to settle.
And you say it looks perfect every time. I'll do it a few times. Just watch it okay.
Watch the centroids move like a little comet trail. We, you know, just go and find it and then watch that the circles themselves don't ever move to.
To such a robust algorithm. Such a pleasure to watch. It works every single time.
Knowing all this can fail. Okay. So that is called K-means clustering okay.
So then that means you can find k means meaning you can find k clusters.
So then you can say these are one set of documents. Second centers are for sale.
If a new document lands here then you will call it this class.
Like whatever green will call the new document green. You simply upload the book here and says, can you suggest me more books that are very similar?
You would then return all these books but not return those books. It's obvious.
Okay. Very simple. K-means clustering, and you can go read the code for all of this given.
Let's see how this all works. Just as much fun in the program, click here.
To see like that Euclidean distance. I told you this.
I think last class before last and I before, I was aware that this all uses simple distance measures, right?
That is how it knows that, you know, these are all like one cluster in the cluster. Some people even question things like that.
You know, they think, why should it be Euclidean? Then you have this new thing called manifold distance.
So manifold distance is non-Euclidean. There might be better calculators okay.
Likewise people actually question cosine similarity.
You might have better alternatives. I'm yet to give you. I have yet to give you links for those.
But let's do it a few more times. Thinking more. Cool. So then we can use this for clustering.
K-means clustering is one of the best clustering algorithms.
So now you understand okay. And then I'm going to go in the back and then show you like more cool things that are happening.
Especially in this last part where I show you. Okay I talked about l am operating system okay.
So some people actually think that. Oh yeah.
By the way Microsoft pretty soon I think one of all developers conference I think they're going to show the world something called AI PC.
They already have air acceleration chips that the one I and I had a pair of
windows laptops and then Bing search all that will be actually accelerated.
Okay. So that's pretty amazing. This already happening. So Microsoft Microsoft Air PC, it's such an interesting thing to even say.
Uh, not these not these. Yeah, exactly.
Three weeks ago. Might as well look at it. Right. These are words to work.
And then I'll get back to what you're saying. Solutions. But operating system.
That's what all of this. Let's start with that. News from Microsoft Crossing just moments ago.
Our Steve Kovach is at CNBC headquarters with those headlines.
Steve over here. Hey there Frank. Yeah. Microsoft right now launching its first artificial intelligence PCs.
Now these are updates to surface computers. The Surface Pro ten is the new one and the Surface Laptop six.
But what makes it different and really what it is an IPC.
Well, it's a term you can expect to hear a lot this year. And here's what Microsoft says its new IPC can do.
They have a special kind of chip inside called a neural processing unit, or NPU, that can support GPU memory patiently.
Now, in Microsoft's case, that means Copilot. Of course, that's a digital assistant that's now built into Windows 11 for business users.
And Microsoft sells businesses the Copilot service for a whopping $30 per user per month.
So that's pretty cool, right? That all the copilot functions can be accelerated.
When you say make me new image in our compose a piece of music for me, it'll happen in much faster time than if you don't have it.
So they hope to launch like a whole bunch. I hope to sell a whole bunch of new surface PCs, right?

But Apple has a very similar thing, you know, also in the making.
So I think all the all the big, you know, um, laptop companies,
operating system companies would do it and even meta, even Amazon, they all have what is called an inference chip.
An inference chip is simply a custom designed processor and is meant for lambs and other kinds of inferencing.
So these are all fascinating right along those lines. Then this notion of a, uh, AI operating system.
So that is what I was going with this. I always read this little you see here, you have the actual OS, right?
But then there is the kernel. And so for example, tool manager, what the [INAUDIBLE] is a tool?
What is an Asian agent is a small mini prompt that I told you so,
rather than one giant prompt that does five different things, you write five mini prompts.
Each prompt is called an agent, and each problem can even be told to go outside the system and run something external.
For example, run a mathematical call to compute the thousand you know, digits of pi and average a thousand digits.
Then those are called tools. Okay, so the idea is that agents can be used made used to use can be made to use external tools.
So once you have external tools you open the whole thing up to anything you can do SQL database search.
You know, you can do image processing and find some kind of an output between it.
So there's no limit to what agent can do.
So you no longer have to worry about what the actual LRM knows because you can make it go out certain decisions or this many problems.
And how do you then chain a bunch of them? Like how do you make one agent use one tool and get the result and pass it to a second agent as an input?
You need to chain them. So that is where the whole prompt chaining comes in.
And languages lang chain. So lang changes.
For example the JavaScript version. If you want to know about all of those, you have to keep going to deep learning that I.
So this is the very best site that continually, even last week to put out something brand new about serverless.
Um, rag rag stands for a trivial augmentation. So the one there is a serverless course just last week.
You can actually go in here, right? Let's see. Um, yeah.
So red teaming, that's all about, like, testing. This is very cool. Um, yeah.
See, like this. Right? One is in many words, it's all about agent programing.
Yeah. Agent design patterns is suddenly everything is moving to actually agent,
which is actually great because agents are how we humans can change agents together rather than let the machine do it all itself.
Okay. Things like Devon would be a good example. So when do we call all of those?
They try and do it all themselves, but then they might actually fail some more.
Uh, better, you know, more efficient approach is actually for us to do it ourselves.
So look at that one. Like my index is also a different chain programing language and agent.
Same programing language. So there's like 2 or 3 agent chain programing languages like my index would be one of them.
And the lang changes another one of them Lang changes is actually over here.
Um, yeah. I'll tell you about vector databases pretty soon.
It does even lamb ops, those functions that are becoming so standardized about what agent can do.
So just like you have DevOps, your data ops, your MLOps, and now you have lamb ops,
you know, targeting like again like I said, more and more well understood.
And then like a role that other people. Yeah that one that is highly worth knowing as well.
Again you chain them okay. Cool. Like here like this.
Orchestrate and chain different modules together. Because we orchestrate we know what we want.
And so each little piece is somewhat intelligent but not super intelligent.
So we know that this whole thing will work. Whereas if you let it chain itself, it might fail at the very end.
So we inject human intelligence into it, which is the reason why it will actually work.
Okay, cool. So there's a lot here okay. I want to tell you, like some live music things pretty soon.
Yeah. So my point was about the whole, uh, you know, a operating system, as they call it, which is basically an operating system.

Cool. That's useful idea, right? All of this is available, by the way.
You actually can try it. Okay. You can try to see what this even means.
These are small libraries. These are small language models.
It's like a mixture. All in all, a gamma small in the sense file size wise.
Maybe the ten gig 15 gig maximum. It means in less than 20 minutes, ten minutes.
Your laptop can download all of that and put it in your hard drive, and you can locally run it and you can actually see how cool it is.
Oh, okay. By the way, you know, I mean, I can go off in so many directions and I'll go off in one more direction.
This is purely mind blowing because Andre Karpathy is a pretty famous name in machine learning.
He has worked at, I think, you know, like over AI and many other places. Okay.
So just yesterday I think he released this crazy cool thing that is called l l m l l m dot c.
But there is no file called LMC. Does that train underscore gpt2 GPT two?
That's the I'll show it to you pretty soon. Imagine a C program that is about 1160 or something.
Lines of code. You can 100% understand it. It has like pointers.
It has pound defines it as function calls no classes, no object orientation nothing.
And it does entirely train the GPT two architecture.
So a few years ago, before GPT three, all that happened, right? OpenAI made a huge noise about gpt2.
They told the world, this is so dangerous, we think that it can go wrong and not release it fully.
They made a big deal about it. Okay. But now it's like commonly understood, like simple knowledge.
Anyway, so he Andre Karpathy takes this architecture and takes a large piece of like Shakespeare text file.
But the best part is you can separately run a file you want and trains that GPT two entirely from scratch using C, okay.
It means you can fully understand it and it compiles and run so fast it is pure c d okay.
Every one of us, including me, should run this. Then you all know what the attention mechanism is.
You know what? Um, for example, uh, layer norm is, you know, you know, you know about self-attention, you know, about transformers and over encoders.
It's all the C code sitting there for you. Okay.
Look at this, Andre. Actually, let's go to GitHub directly github.com.
Slash Karpathy. Everything that this guy does is pretty cool, so you can check it out.
Whoa, Karpathy. Lemme see.
Four hours ago, and that's just making changes as we speak.
I mean, this thing is super amazing. See that? Hello.
I'm training in pure see, and if you have a slightly accelerated laptop that has some kind of GPU, it will use Cuda.
Otherwise it is willing to run on pure, you know, CPU. Okay, you don't need anything at all.
You don't need anything. See that single file thousand lines of code is actually up here.
Uh, train GPT, it's in here, so we'll go there in a minute.
But look how neat this is. Okay, so, uh, he's going to add more things.
So this is the preprocessor, meaning that the Python script, it just simply reads a text file that he has in this directory and then tokenize it.
You know, token means roughly words. Okay. See that one. So we take that text file.
It means what OpenAI did was rather than this,
they used exabytes and petabytes full of English words that I found, but in principle the same thing as this.
Okay, so imagine somebody give you English text, in this case Shakespeare.
You tokenize the heck out of it, because the tokens or what?
I can be calculated by the prompt, say, if you give it a prompt, tell me about, you know, Shakespeare's lovers,
then suddenly start answering you because it knows what words each word is called a token.
Okay, so it's doing token prediction, but that text file is being processed by this very simple little Python script.
So tiny. Write them and they use a transformer that already exists.
Okay. For the tokenizing okay. Not quite there. Um, it's all up here.
I forget why it is okay, so it's not writing the actual tokenizer himself.
It's like a text somewhere in here. Text replace Np.array.
Yeah. This one. Uh, tokens p file name.
I have to go and see why it's happening. In fact, that is the output.
So as I tell you, after the tokenizing is done, there's an output that is created of the tokens.

It's a bin file serial binary file. And that file will then be used, you know, for fine tuning.
See, we still are not at a point when we can train this entire thing from scratch.
So we do a slightly, uh, simplified version of training.
So what is this training that we talk about? And you have a testing.
Okay. Oh, here it is. Train GPT two t GPT two that see this all Cuda.
So if you have Cuda if you have a GPU the training can be much faster.
It's a coder compiler. Cuda means you do GPU programing, but not for every single core manually.
By writing a for loop for each core, you pretend you only have one core.
You write the code that is called code. Then we Cuda compiler will take the one code.
One uh, core code that you wrote. Imagine you have many cores, meaning many parallel processors, hundreds of them.
You pretend you only have one of them, you write code for it.
What the compiler would do is make that code run in parallel on all of them, so will dramatically become 100 times seven times faster.
That is what NBCC does. Otherwise it's all like, you know what we call templated C plus plus C like metaprogramming.
It looks like C plus plus, but it's decorated okay. And this Cuda compiler would under loop unroll all the loops if necessary.
But then the the the goal here is actually that one trained Jupiter to that C1116 lines of code.
I said 1160. Well that that's all good.
You will not see a smaller Llvm training code, especially in C.
Just wouldn't say this one is like poetry. So if you want to see very well written code so please read it.
Okay, in here is everything. Suppose I say to you transformer architecture, which I'll show you next time.
And I still want to tell you myself how it works. If I say transformer architecture, suppose I'm going to say, uh, lamb.
So we don't look at transfer robots, then what's going to happen? You see diagrams like this over and over again.
Okay. All these now parallel layer self-attention. You know what, right?
We'll go ahead and understand what is all happening. This code will explain all of it to you.
There's no magic. There's no second file, there's no library. There's no encode file, nothing pure.
See, if you understand all this for your life, you understand how this brand new radical architecture works.
Okay? It's just beautiful.
I intend to, you know, like I said, I mean, I know some of, like how it works, but I want I can go through it line by line and folger's get it.
Okay. It is great. It's all in here. And it's exactly what verse one E and Nicky Palmer and the people from USC, which are dead on there road.
This paper call attention is all you need. Look at what is included.
Basically nothing on OMP multi parallel multiprocessing in our library.
So you don't even need the amp part. You can delete it okay. It means it does not depend on anything.
It's a pure C program. Wow. Amazing.
Okay. So please read that. And coming back here.
What else do I have to show you? Oh, yeah. This one is actually sad because when you talk about searching or viewing.
Really? All right. So YouTube, you know, had some videos and about fintech and bitcoin and all that.
Right. Many people watched it. And then the US government basically said to Google, hand over the and um, basically the identity of people that actually watched it, that is like super scary.
So read this. Wow. Like this.
Names, addresses, telephone numbers, and other activity. Certain YouTube videos follow.
It means you might watch something tonight that is highly know, innocuous. There's nothing wrong with it.
But five years from now, somebody would decide, wow, that was bad.
Nobody should have watched it or watched it. You'd be in trouble. Okay. That's bad.
You know him like this? You know, the whole money laundering, you know, some shady thing that's going on.
But, you know, they're just basically public videos, okay? Come on. You don't have to go out of your way to watch it.
They're sitting there. And then the government wants to know.
Watch it. Come on. Right.
So yeah, this is what Google tells you.
We examined every single case, pushed back, you know. But the government can also push back even harder.
All right. So then. A bulk, you know, bulk warrant.

How? This is what we're worried about, okay? Mapping software for drones.

You know, there wasn't really one. You would watch it for machine learning purposes.

You know how drone. You know, mapping works, and suddenly somebody thinks you should know it.

All right. Okay. That, uh.

That's okay. Okay. Check this out.

So on a more fun note, I'm going to show you three things which are supremely useful to musicians electronic musicians.

It's all about clustering and classification. What can you say?

If you make EDM, one of the things you will have is this thing called a sample library.

You want a certain kind of a drum? I want that exact crash cymbal sound I'm looking for.

I might have a folder called Cymbal Sounds. I might have 1000.

MP3 files are WAV files. They're all slightly different.

I can play them, you know. They're all different.

Eventually. Pick one of them, right. So we start collecting samples throughout our lives.

Okay. There are some people that might have up to, like, 500,000 or even 1 million sample files, and they're all named a certain way.

They're sitting in directories. Right? There's no good way to go through all of them.

Usually we get bored and picked the very first file name. Okay. It means, why the [INAUDIBLE] did you and pick all of those.

Okay. There's no good solution, right? We're just a file name. But here comes clustering.

So what's this? Super cool similarity search on music producers.

If you struggle finding sounds and samples that work for you, then you have to check out.

This plugin is called arcade and it's my number one go to for building.

Everyone. This is about matrix. And today I just want to talk briefly about the sample management tools or sample librarians.

See that's what the files look like, right? It's I mean it's mind numbing because it's called club.

You know gun action club eraser, one eraser to look what the distance between 1 and 2.

We don't know until we play them. Okay. The numbers don't tell us anything, but you can cluster them based on what is actually in those sound files.

That is more amazing. Okay, so you look at this, then you can visually navigate and click on a sample library,

almost like the k means will mean that I showed you 36,000 samples.

What the [INAUDIBLE] you said has $100 spread across quite a few directories.

And this is common for it looks like I think a lot of us terabytes worth of samples were collected from all over.

We wouldn't trade samples to each other. And I have my manually curated sample library here.

There's drums that guitarist and I use that makes sense to me, right?

It just goes on folders and folders. Sponsored content.

How are you going to find the one magic tone that you want? Uh, which have their own sample folders inside them that I can't really mess with.

The structure of all my native instruments are going to go because that's those.

That's what they look like, right? And then you can click on the metal plant. Now there are a few different sample managers.

Okay. The program required called synonym next synonym.

But so on and so on.

An image a commercial piece of software you can buy which will go in every single file audio file and analyze the actual audio samples.

You might call it content based filtering. Okay. Content based recommendation.

It'll actually look at the like for example, fast EDM, slow jazz, male vocalist or female vocalist.

Will then cluster them based on what is actually in the file, not the file name.

Okay. And then it's going to visually show it. That's great. It's been around for about a year.

There's one called a guerra, but there's a there's a trick to this.

Right. Okay, we're gonna ignore all this and play the and run the synonym program in their kick folder.

Right. They're still displaying files that are kicks, but they're not named with the actual word.

See, that's the thing, right? By word basis. Those searches on the right hand side.

But unless you call it the right thing, you'll never find your actual file.
Okay. Slightly different, right?
They they design. But then we approach the problem. So it always comes.
Oh my God let me do. And my teacher my producer tutorials.
So you'll find where I just I have an hour long.
This is amazing. So each is a wave file. That's my favorite.
And wave files that are sound slightly different. There are next to each others.
Oh, aren't specifically about drums. This is a drum oriented kind of sound.
Oh my god, some things you'll use for cluster in it, but you know, for anything outside of drums up, you know,
I need a good sample manager and right off the bat song and it brings a kind of value that the others different.
So with all your knowledge about, you know, clustering and things. Right. This second useful strategy you should write.
You can do this for video and go to NFL. Go to big, you know ESPN, all these sports companies and sell your product.
Tell them if you upload some kind of a quarterback you know like you know doing like a field goal.
And then I'll be able to pull very similar clips based on this kind of similarity.
So they would love you because right now it's all called damn.
It's called digital asset management. So digital asset management is what many studios use to, you know, catalog clips.
But they're all text and keyword based. It is a old way to do it. Okay.
So the keywords are not that good luck to you. But now you can analyze video frame by frame and just imagine how amazing that is.
Football games played in heavy rain and sunny weather.
It automatically determines for you. That's a very quiet.
I'm going to give you an explosion and it's accurate, right?
That sounds like thunder in the distance. So the point is there's no where.
None of the dissonant the sounds are. How?
No. All of them. That's why I'm going back to base kick. You can click directly on it.
And if I go to a different talk here, there's things in comics, by the way.
There's a similarity. So you can actually set similarity in a common similarity you want right now just very someone's compilations and they're,
they're actually occurring and zooming through my small great brain.
And then every dot is actually a file in the folder so far purely text based search for base.
And suddenly. Yes. Because this space cloud is organizing all the sounds, and there is some kind of fuzzy barrier between these two things, right?
Temporal matter. So all the kicks that sound similar to each other are clustered together.
And that is the whole question these are going to be. By the way this imitation is using our terminology.
Hopefully these are going to be more energetic. Next these are going to be more thumping kicks and so on.
It's kind of like the tramp. So that's one of them, right of one. Then to that it's you can use machine learning to actually train these things.
Okay. In other words, neural network. Well, what? I have no idea what I launched.
Oh, sorry. Oh.
Neat, right? Okay, so one way to do this is to not use what you today call a neural network.
You could do it the old fashioned way, which is pretty much analyze sound samples, okay.
And look for similarities. But then you might actually inject some musical intelligence that you have into it.
Pandora had a project like that is called the music DNA project.
It's a music DNA project that took many thousands of actual songs throughout their Music Genome Project, Ryan and Pandora.
And then they classified every single song, every single piece of music,
instrumental or vocal, using about 100 different parameters that musicians would identify.
So it's a big table, you know, song name, 100 columns.
And what is the value for each column? So then you can obviously cluster based on similarities of those columns, right.
That is what this is is still an amazing thing by the way. You can actually get your data okay.
I can say music, DNA, project data. You can download all this data for free.
By the way. It is super cool. Pandora community uh, music genome project.
So you can go look at all this, but what I'm going to show you is something more recent.

Okay. So the select pretty neat. Think of this as the older way to do it.
But then what else do we have this one. Yeah.
One more lens. Mr. blue sky.
Over and out. You're just not comparing.
XL by excellent audio and Atlas two by Algunos.
Both of these are absolutely fantastic beat making tools.
And so two different programs right. Excel versus Atlas to want to go back to regular sample browser.
And as a quick disclaimer, I was sent free samples of each of these 31 labels that it says claps snare.
It's not as high handed, and I'm being honest, raising my competitors in the directory.
So Maxo and Atlas two have their own strengths and weaknesses.
Exo excels in its interface. This is just.
But personally,
I've sat less for a little bit longer than Atlas two because I know enough about Folsom that I can completely see what is clustering, right?
It clearly just works on the sound and the sound samples and then into a map.
So this is a neat comparison between the two version, which I thought was a different option.
Anyway, this is all exciting because it's how we take what we know.
And this group, you can go in here and you can then press start Randomize Indian music, for instance.
You know, it's called a South Indian North Indian version, but they both are, but basically the same.
They use a very evolved version of a scale. It's called Rogue Dragon.
So then you can cluster things based on rocks and those are amazing.
Okay. Then you can quickly find all the songs that sound the same.
The same rock. Okay. The last one is this one coalescence.
Oh. Check it out.
Neural computing. Native models have simpler neural networks to organize, analyze, and playback sample slices.
How cool is that? So now this one uses like what you would call today's air.
Okay. This is a very short video summary of a plan.
These are new useful markets.
Okay, that people actually will use mentions craving conflicts rather than write the same application over and over again.
That's been like the green precious for you.
So you can move that around and then it actually picks things that are basically close to it.
So running inside a host program called Ableton Live.
So for us people that are programmatic, right, more mathematically know Ableton Live is the perfect music program to learn.
For standard classical musicians trained on guitars. Ableton live drives from crazy for people like, I totally get it.
And then something called Max. Max is actually like this.
In Max you have max blocks, you wire them. So that is what called a max for live instrument.
In other words, this architecture was used to program that and this Max is now running inside Ableton.
Max used to be a separate company called cycling 74 and Max was until it bought by the Ableton people.
So now it's simply Max. There's no more Max anymore. It's Max for life.
It's a max for live can have plugins that called Max for live instruments.
So that is one of the instruments. You so cool right. So you like this?
Build your own instruments and effects. So this person that I showed you said I'll write a neural network based music classifier.
See like this. You can actually wire max like nodes together. So node based okay.
It's randomly entertaining today. That's my point. And then we'll get back to our new topics over on this.
Let's look at how cool Max can be. Now running in Ableton okay.
That's a max device.
You can do crazy cool things with your Arduino and motors and, you know, LEDs, light different and resistors and turn them all into sound.
So programmers playground. Huh?
You and your dog can make music. Anyway, that's just one example.
That's not. Wow.

Okay, so I love a cause, apparently. Great.
So then what else should we do? I'm just having so much fun with you guys.
All right. So now maybe we can actually go click on the lectures.
Okay. And just look at like I said, random topics but still all somehow related to things like search and alarms.
And then over here, too far from what we you need to look at. And there's no rhyme or reason to like why I picked this, okay?
Just like, you know, you can't ask that important.
There's no. So what I'm trying to do is leave some of the old stuff that we looked at behind, and then just jump you straight into bleeding edge.
And bleeding edge, by definition, is so rough, so fractal. There's no good introductory starting point.
Everybody start here. It's not like that, okay? Just pick whatever the [INAUDIBLE] you want to jump right in.
And before you know it, you'll be like part of it.
So I'm going to show you, um, in the last two years, like I said, it changed, like, so much, and nobody's jumping in, right?
So we'll just look at a bunch of different things. For example, today I can tell you about vector databases.
Each one is a very big field, even just this one thing alone.
Thankfully, there's nothing new about vectors themselves vectors, vectors, vectors.
But where they're all available, it's actually brand new.
In other words, every single day I come across a brand new, uh,
existing database that has nothing to do with vectors at all, but suddenly they announce vector support.
It looks like the entire universe wants vectors. Okay.
And the entirely pure play vector database companies built from scratch did nothing but vectors.
So a little bit about okay, because vectors are very important.
Vectors are one of those things that actually, like I said, they make algorithms look very good.
So definitely spend so much time learning about vectors.
And they're so easy because many vector databases are simply Pip installs.
You know, many are just Python scripts. You can just go read to see how it works.
And so then again, the more you know, the better. It is about vectors. Okay? Vectors are how you make similarity.
Cool. So I want to get back to ImageNet and tell you a little bit more about ImageNet.
I mentioned random things about ImageNet here and there.
But ImageNet still turns out to be a pretty valuable resource because it has many million like ten or so million images.
First of all, you know, painstakingly calculated, um, collected by humans and also properly,
hierarchically classified by humans and labeled by humans.
So if you want good ground truth, you know, how good is my AI?
You still need something very valuable like this. And there's nothing bigger than this.
Okay, by the way, we should not rush in alarm to add to this because LLN might hallucinate.
Meaning add bad facts to this and pollute this entirely.
So you want to keep a clean well full of good clean water. So ImageNet is one of those.
Okay. So let me then tell you about ImageNet what I didn't tell you before.
And I can keep going. So you can go to image-net.org.
Very interesting. So it's a hierarchy. First of all ImageNet.
And the ImageNet is based on the WordNet hierarchy. So definitely told you about WordNet.
In WordNet we take the English language for example I'm gonna say things like animals you know that's a big hierarchy called animals.
Under animals they might be primates and under primates they might be apes under apes.
Apes might be different kind of monkeys. And finally, you know, the human apes, okay, because the whole state hierarchy.
Right. But we know from English what those words mean. So we made the WordNet hierarchy.
Likewise, the ImageNet hierarchy would then for each of those words in WordNet,
go find a bunch of images and then you know that they are hierarchical, organized.
One thing that many, many, many machine learning programs cannot do well even today.
In fact, lm also the biggest problem. You know, the only problem really, you know, in a way it is all this idea called generalization,
generalized generalized, generalized, generalized, generalized. Almost no program seems capable of generalizing.

If you teach them something very specific, like what does a wallet look like? What is chocolate like water bottle? Whatever it classifies like crazy, you know it can do a billion times faster than you, but then it cannot understand that a water bottle and a cup,
and maybe a juice bottle that looks like a rectangle or all used to hold liquids that humans drink, that is a generalization.
Okay. So that means if you give it like a cardboard box,
it might not occur to it to actually put it in its mouth and drink, even though it seems to be filled with liquid.
Nobody told it that, but a little kid would get it right. A kid is used to, you know, sippy cups, right?
And then like cups and water bottles and a little grapefruit, you a little punch that comes in the cardboard bottle.
No problem. Just start sipping it. Why? Well, I'll tell you why, but you might not like my answer.
Because we have a body. Okay, I tell you that a lot of times. But then what?
I would try to fake, like what a kid would know, but without the body.
We call that generalization fail over and over and over. Anyway, it's all about generalization.
So if I give it in a WordNet or ImageNet, even a picture of like a human,
you can generalize that all the way up to say that's an animal obviously would never call me a tree, right?
It's a simple example of generalization. In fact, babies are able to generalize about things that are not even seen before.
They've never seen something, but they can still generalization. It drives their people crazy.
Okay. Anyway, so then that's about ImageNet. I'm going to tell you more though.
So then how can ImageNet be used for things like vision transformer.
So transformer is is a radical idea where words in a sequence of words like sentence of words.
Pay attention to each other. Every word. Pays attention to all the words that came before.
That's why it's call. Attention is all you need. But a word is a sentence itself.
A one dimensional string of words, right? An image is not one dimensional.
Image is two dimensional. But it is still possible to break an image up in small little patches, like small sub rectangles of images,
and how each patch pay attention to all the other patches in the image.
So then you have something called a vision transformer.
In other words, what was invented initially for purely one day has been entirely adapted to work in 2D, and it works amazingly well.
It's actually better than what's called a CNN.
So CNN used to be the state of the art for image classification, but Vision Transformer already better than CNN.
So then I want to tell you how ImageNet can actually help you. Okay. Again, you can learn vision transformers versus CNNs.
Okay. Okay. So then you know overall it's more robust okay.
It's like lots of different right there. So the same idea of encoding is no used for an image patch decision.
So if we break an image up into nine pieces then each piece can pay attention to all the other eight pieces.
It's called quadratic attention because as n squared and every I would have to pay attention to order in a not I.
All right. Cool. So all that aside, where are we. Okay.
So like here, you know, if every patch is able to, you know, pay attention to all the other patches.
Among other things, the machine can learn that the dog is only made of these dog parts.
That is not part of the dog with CNN that there was actually a problem.
If you take a whole bunch of eagles that are all against the sky and you call the whole thing an eagle,
the network learns that the network also thinks that the blue sky behind the eagle is also part of the eagle.
Then if the eagle is lying on the ground, you ask, what bird is it? It might call it a squirrel or something, right?
Completely fails because it doesn't know where the squirrel, the eagle ends.
Where the sky begins. But with vision transformers, it's a lot better.
It's more robust. Okay. It makes the CNN problem basically go away.
All right. Cool. So then how would like imagine that help with all of this so we can go look at it for a minute.
Trying to see where we are. Yeah.
So let's go here. Okay. Cool.
I'll go a little faster than the one player. Like two minutes for you. What is your.
This is the inventory that by the way. She's now a machine learning God specially.
Let me show you something. They actually won ImageNet. Okay. Okay, so they are the reason why modern deep learning was born.

The boys betting the ultimate. Probably a daughter.
People are going on an airplane. Those are the people that are going on an airplane.
This is a three year old child describing what she sees in a series of photos.
She might still have a lot to learn about this world, but she's already an expert at one very important task to make sense of what she sees.
Our society is more technologically advanced than ever.
We send people to the moon. We make phones that talk to us, or customized radio stations that can play only music we like.
Yet our most advanced machines and computers still struggle at this task.
So I'm here today to give you a progress report on the latest advances in our research in computer vision.
One of the most frontier and potentially revolutionary technology in computer science.
Yes, we have prototype cars that can drive by themselves, but without smart vision,
they cannot really tell the difference between a crumple paper bag on the road, which can be run over, and a rock that size which should be avoided.
We have made fabulous megapixel cameras, but we have not deliver sight to the blind.
Drones can fly over massive land and not enough vision technology can help us to track the changes of the rainforests.
Security cameras are everywhere, but they do not alert us when a child is drowning in a swimming pool.
Photos and videos are becoming an integral part of global life.
They're being generated at a pace that's far beyond any human or teams of human could hope to view.
And you and I are contributing to that at this TEDx.
Yet our most advanced software are still struggling at understanding and managing this enormous content.
So, in other words, collectively, as a society, we're very much blind because our smartest machines are still blind.
Why is it so hard? You might ask? Cameras can take pictures like this by converting lights into a two dimensional pixels of numbers.
Teach computers to see. Our research field is called computer vision and machine learning.
It's part of the general field of artificial intelligence classification.
So ultimately we want to teach the machines to look at that just like we do.
Naming. All because a neural network has been trained on many million pictures of cakes, of chairs, of kids faces, t shirts, chairs, trees, whatever.
Right? And then you train it over and over and over and over again until it minimizes the error in classification,
meaning the neural network numbers call weights. They learn, so to speak, meaning they calculate the proper numbers.
You need massive piles of geometry of things, understanding relations, emotions, actions, and intentions.
You and I weave together entire stories of people, places and things.
The moment we lay our gaze on them. The first step towards this goal is to teach a computer to see objects.
The building block of the visual world in itself is hard to see, right?
You try and give the picture of a cat to a neural network and it might say dog, did you get a picture of a bird?
And then it makes a chair. That's called misclassification.
Then you have to go backwards and change all the numbers that made the misclassification happen in the first place, and try in the forward direction again. So keep doing this back and forth.
Back and forth. Back propagation. Back. Propagate the error forward propagate the test and see how it's training.
So finally there comes a time when it's like 99% for example accurate and how it classifies.
Then comes the real test. Give it a brand new cat that the system has never seen before.
And thankfully, hopefully because of all the training that went on, you will take your cat and correctly say cat.
Okay, we can feed some cat treats. So the on the road it drives on roads that has never been driven on the car,
never draw on and still identify stop signs and traffic lights and pedestrians properly so nobody's killed.
Okay, so it's almost like a magical thing, but it needs lots and lots and lots and lots and lots of data to learn from these training images.
That's where ImageNet comes in, okay. After all, a cat is just a collection of shapes and colors.
So then the one thing that neuron that we don't program directly, we used to do this okay.
In computer vision we still do it. We think that a cat is a cat because it has a triangular nose.
It has long whiskers that stick out this way, and then the ears that go up like an upward control conjugate.
That is an explicit geometry specification on our part.
It might work nicely until the catalyst its head and then goes this way and suddenly the classification fails.

Okay, so we cannot put an if statements. Let's say if the radius of the eye is less than three in order, right?
If the triangle is like a one inch from that, that is, it's untenable.
We cannot scale the whole world like this. So then instead, we don't do statements, we use data.
We use many, many, many, many things that look like cats and let the neural network somehow figure out what makes a cat a cat.
We don't explicitly specify it, but we punish it when it calls a cat a dog until it adjusts itself.
Okay. That's all. And this is what we did in the early days of object modeling.
We industrial computer vision still works like this, and it works pretty well.
Circuit boards, phones, Apple's face, a chubby body, two pointy years and a long time.
You make it look for the geometry and it works. But in the real world it cannot.
Let's cut. Yeah. So what are you going to do now? Right now you have to add another shape, a viewpoint to the object model.
There's no end to it. What if cuts are hidden? What about these silly cats?
Huh? Cat. Jesus. Now you get my point.
Even some I reason as a household pet can present an infinite number of variations to the object model.
And that's just one object. Okay.
Imagine that. Okay. Child would have seen hundreds of millions of pictures of the real world.
That's a lot of training examples. So instead of focusing on solely on better and better algorithms,
my insight was to give the algorithms the kind of training data that a child was given through experiences that project.
Oh look, imagine that ever created. We downloaded nearly a billion images.
I use the crowdsourcing that I saw.
They got random people in the world to label them, to help us to label these images $0.01 per image for labeling images that are better.
The biggest employers of the Amazon. Look at India.
Oh my God. You know the big English speaking countries okay.
And then hire like workers, put the headphones on, just classify label them.
Countries around the world helped us to clean sought a label.
Nearly a billion candidate images. That's crazy.
That was how much effort it took to capture even a fraction of the imageries.
A child's mind takes in in the early developmental years.
In hindsight, this idea of using big data to train computer algorithms may seem obvious now, but back in 2007 it was not so obvious.
Look, I'm going to go here and image that after all, that's how I found out through my college years.
So we carried on. Look at that nine image.
That project delivered a database of 15 million images across 22,000 classes of objects and things organized by.
That's WordNet in both quantity and quality.
This was an unprecedented scale. As an example, in the case of cats, we have more than 62,000 cats.
That's all. All kinds of look, we need more cats. Questions?
Cats are cool. And, uh, across all species of domestic and wild cats.
We were thrilled to have put together an image that I wanted the whole research world to benefit from it.
So in a Ted Fashion Week, that's pretty cool. So you today can actually download all of Netflix community for free.
If you ever tried asking questions to ChatGPT for research.
You would agree that you cannot trust it. Liner is difficult for most a person like ChatGPT.
I will quickly get past this so we can talk about other things as well. But you mentioned that is still fascinating for all these reasons.
You mentioned academic papers. So what happens to sites? Okay, so check this out because imagine it's cool.
Machine learning algorithms called convolutional neural networks pioneered by Kuhn Akiko Fukushima.
So now you might call CNN is obsolete because of vision Transformers.
But back then, CNN with many, many, many layers was what was used to win ImageNet.
In other words, here's what happened. This is true. ImageNet was out there as a publicly available data set.
Anybody in the world can download it and create any neural network architecture that think is pretty cool.
And then the real test comes when you take random pictures from ImageNet or even categories of pictures in ImageNet, but not in the database.

And see how good your training classifier sorry, not training, how big your classification accuracy is for the longest time, meaning for many, many years the accuracy was in the low 20% and so on.
Okay. Pretty pathetic and nothing to write home about.
Imagine that suddenly because they use like a deep learning CNN architecture, the accuracy went up by a much, much, much higher value.
I forget what it is. We can Google it. But then at that there was a watershed moment.
Oh my God. Now we can actually classify ImageNet images to a pretty high level of accuracy.
And that is when the new deep learning revolution was actually born. Google.
That's fine okay. And and but I wanted to tell you that even this is not the state of the art anymore.
It's not CNN just even oh, actually trying to get here on network.
So all that one neuron does is it gets inputs, possibly from other neurons.
Its job is to learn something called a weight, a multiplier, a weight, a weight.
Are we starting points? Right.
Then what the floating points would do is after the weights, I calculated this x one times w one plus x two times w2 plus x three times w3.
That is all the whole neural network training is all about. What do you think?
W1 w2, w3 are initially completely random Python numbers between -1 and 1 entirely initially random values.
Okay. So then when I show you a picture of a cat, imagine this picture of a cat.
Okay, you entirely multiply by random weird numbers. You have no idea what you are doing.
Then you take the result. You pass it through a curve called a sigmoid curve.
That is to make it nonlinear. Okay. In the end, the result might be turned into a number.
You give it a cat, right? It says Frisbee. Like what the [INAUDIBLE]?
Completely wrong. Right? This is not the only neuron. There are many neurons.
And so they all have their own weights. Okay. So together they all did not learn properly because everything is random.
So a cat got classified into a Frisbee. A person got classified into a parrot.
That's a good thing. So we use many thousands, if not billions of training data.
And what label what. And so we expect when we have the wrong answer from the system, we go backwards from this layer to that layer.
Each layer is made of neurons. Okay. And make these weights change.
You know how I said that I weights, right? You miss the errors gradient and make the weights be slightly different compared to what it used to be.
Slightly learn a little bit, then in the forward direction. If I pass all those images again, I won't be as bad as before.
I would be slightly better. I used to be only 20% accurate.
Now I'm 22% accurate. Still not enough. Okay then.
Still back. Propagate. Change all the way to one more time. Try again.
Now the accuracy 130% 31, 32, 33, 95, 98 ohmygod.
So piece by piece by piece, forward, backward, forward, backwards. So she probably can talk about that.
Okay. That is all training is. It takes input from other nodes and sends output to others.
Moreover. But the weights it's all about the weights or even millions of nodes are organized in hierarchy.
Deep learning means lots of layers of layers to the brain.
In a typical year, I'm going to sharply disagree with professor really and said it's all not true at all.
A brain is infinitely, horribly more complicated okay than any of this very cheap Python functions.
Okay, so please don't even get me started. Not that you are, but she's wrong, okay?
That is the reason why we still don't have, like, you know, brain like we used to train our object recognition model.
It has 24 million nodes, 140 million parameters.
So parameters mean all of these, like, you know, take a function call.
Right. Every function has a bunch of inputs that function. That being said, what if you add all the function inputs together?
We call them parameters. So OpenAI you know, ChatGPT 4.5, all of that.
There are parameters also same kind of parameters. Um, OpenAI 4.51 trillion parameters.
Some of the Chinese LMS like 2 trillion parameters.
So parameters are mind bogglingly big. But some people actually think that the era of these large language models getting larger and larger is over.

Instead, you go the other direction and say, what about as little as 7 billion parameters?
What about 2 billion parameters? It looks so cute and tiny.
What about one point million parameters? By the way, GPT two, when you train it at home tonight, it's about 120 million parameters, okay.
And it still does something useful. So people actually think the better direction to go is to see what the smaller models can do anyway.
That is what a parameter is. You simply add up all the inputs okay. Billion connections.
That's an enormous model. Okay. It became the winning.
Oh, wait, wait. Let's actually go to kitchen. Powered by the massive data.
So then they built this deep learning thing that she talked about and then, you know, train that on ImageNet and then give it new training, new data.
Right. And it won the competition ImageNet and the modern CPUs and GPUs to train such a humongous model.
The convolutional neural network blossomed in a way that no one expected.
It became the winning architecture to generate exciting new results in object.
So this was, like I said, a transformative moment in the history, telling us this picture contains a cat and where the cat is, of course.
Look that's Karpathy. You see that?
So here is a computer algorithm telling us the picture contains a boy and a teddy bear, a dog, a person, and a smile.
That's more useful, right? In a traffic situation. And you want to know all about traffic, not just one thing at a time, like a map.
It's called segmentation.
You segment the whole image into whatever you can segment, and then reason from all of those from the computer is not so confident about what it sees.
That's a weird thing, right? Like a bird was trying to get her to give us a safe answer instead of committing too much, just like we would do.
But other times our computer lets you. Remarkable.
That is pretty crazy. It even knows in actual specific kinds of car model year of the cars.
We apply this algorithm to millions of Google Street View images across hundreds of American cities, and we have learned something really interesting.
First, it confirmed our common wisdom that car prices correlate very well with household incomes.
But surprisingly, car prices also correlates well with crime rates in cities.
Interesting or voting patterns by zip codes.
Wow. The things you find way to beat it is that it has computer.
You can see. You can definitely go watch. I'll just have to order. But I thought you would get a kick out of it.
Okay. She said a bunch of things and I want to tell you some things.
In addition, we have a professor here called Mark Harmon.
He's so famous that in the new shirt department that is being built across the street over there.
He has half a floor and the rest department is all the rest of the floors.
He gets his own half floor close to us because he is a vision pioneer, not vision in the sense of computer vision, actual human vision.
Because if I said we cannot make the blind see, right. So look what he does.
Just one of them should be enough. Okay. Okay, look at this one.
So I thought this girl was working out like crazy, but she wasn't. I met her yesterday at the gym and I asked her, how do you look?
So great. Okay. It's a random non-sequitur Terry and I met.
Okay, so look what he's wearing. Five he already had some night blindness, and he didn't tell me how bad it was.
And we got engaged a few months later and he just kept getting seemed to be getting worse.
And so. My mom had them come to our eye doctor and she said, you know, there's something else wrong.
And so he had a bunch of tests done in near Santa Ana.
So this all the past. Okay. But now look, before we got married, when we got home for our honeymoon, we got the diagnosis and we're still together.
The Argus implant was actually before he went.
It was called Argus, who was conceived way back in 1987.

You know what is happening? This is a monochromatic video camera that's only 64 by 64 pixels.
Little blocky camera. It collects video signals, right.
And the signals go through a wire to a device that is wearing in his pocket.
The device translates those video signals into what the brain can understand in terms
of rods and cones wirelessly transmitted to a chip that is on the back of your eye.
It's called the retinal prosthesis. You have a surgery, and then you have your retina replaced by an artificial right now that is stimulated
by all these waves that the video camera sees you walk directly the person's brain,
okay. And then they can start to make out shapes. That is how the blind can actually see.
It's amazing. My grandmother started to go blind from complications of diabetes,
and it made me start to think about what could be done for patients who are who are blind or who are going blind.
And it gave us this idea of being able to use a computer chip and implanted in the eye to help restore sight to the blind.
The device we're still looking at, the camera here, sends the information to this video processing unit,
and both power and data are then sent back up to this coil, which sends it into the implant.
And so that's the magic part. Blind person wirelessly sent, wirelessly hooked up to this camera in the glasses.
In the beginning when I first got the Argus one, now it's up to Argus two, by the way.
And, uh, told me this is what you kind of expect to see since they block it.
You see, you see where the doorways, lights, somebody's coming towards you.
But your brain is remarkably, remarkably plastic. It learns in a minute.
I'm not in total darkness. I'm seeing a little pinpoint of light.
And I saw it in a way that's pretty cool. And then you have you walking up and down this hallway, either with or without that.
It's like magic in scanning back and forth. And the walls were white, the floor was white, but the doors and the side boards were dark, dark brown.
And when I looked, I could see a vertical line here, lying there.
And I thought, wow, look at this. I could actually go down the middle of this thing,
was out my cane and not walk in anything because I knew I was in the middle and I just kind of hung my head.
And I just about totally lost it because I didn't have to have somebody's shoulder or anything else.
It was me, the cane and that fantastic technology at the time.
And, you know, I've got the second shoe, which allows me to be mobile, more independent.
That's USC Keck, by the way. The device, it gives me more self-esteem, more awareness and and different things in my life than I'd had a long time.
It's really given me a big boost, uh, because I want to be as independent as I can be.
Well, Terry, uh, is an amazing. So fast forward a few more years, right?
You can do this in stereo. You can do it in color. You can do it in eight K and probably power the device using your own blood,
because you can generate electricity from your blood and use it all as self-contained and actually see, like you and I would see.
That's pretty amazing. So anyway, I wanted to tell you that about that and a little bit more about image net.
Okay. Cool. So now we have some slides and I'll go through it a little bit fast.
This one says um how hard is Google to image search okay.
Say upload like an image I can do an image search on Google. Right?
So I just work on the one hand, image search works purely by the text descriptors that we put in.
When we upload an Eiffel Tower picture, it might say Eiffel Tower in Paris or Eiffel Tower, a close up view and amazing tower.
So we use words to search, right? But then, you know, all the usual problems with that.
If the words are not adequate, then we cannot search for the next this all word based.
Then the next step would be even though they call it similarity, it is word similarity.
Okay, that is still mostly the state of the art, but gradually we can actually start to use things like vector embedding where you can
embed all these pictures and they go become clustered at some point in vector space.
And then you image you want to search but also become embedded in the exact same vector space.
Then you can do the similarity search. So we are slowly going there.

But you know, even with, you know, just car descriptions, words, you're able to pull up remarkably some pretty amazing things.
Okay. So it's going to tell you all about that image identification keywords okay.
So tin AI is actually different. Tin is one of the first, um, classifiers that actually uses the images on profile,
meaning the pixels that are in the image still doesn't know what is looking for, but it uses pixels versus pixels.
Yeah, there's still no semantics there, but at least no words. Okay, cool.
Okay. So then. Yeah, like they see the tags. All this.
How do search engines damage indexing? First use tags. Tags about keywords.
Right? But keywords are wrong. Misleading like all that. Okay. Right.
See? Look at that. Use text with images you know, associated with images.
Sunset means it will pull all the sunset pictures. Why?
Because all those pictures you uploaded with all of these, uh, captions that had the word sunset in it, so obviously not going to work, right?
Feature extraction is much, much, much more difficult. This is where things like vision transformer, they can all come in um, embedding.
Vector embedding. So we'll take a little break soon and then we'll talk about embedding.
So these are pixels now okay. So you can somehow extract meaning from the pixels.
That is easier said than done. So the first thing that people tried was well just look at the color of what's in the pixel.
We'll just look at the pixel color. So you have a basket full of apples right.
There's a certain apple red. So if I know what the RGB value was I can go and find in my collection all the other images that have the basically the approximate same RGB and pull it out and hope that it'll match.
So it's basically purely a color based matching absolutely nothing more than that color histogram okay.
It has mixed results. As you can see. We call it correlogram for every pixel in the database with your input image.
The what age classification. You know you've seen all this before okay.
And pull up all the images that are pretty close to the target image that you wanted to search.
So do basically closest match to like hair color layer color layout.
All right. So we have just one image. And just why look.
Why look at one image. Sorry. Why look at one color throughout an image.
Use images. Many different colors. Maybe apples and also oranges nearby.
Maybe we should first segment the image and pull all the orange pieces separate and the red pieces separate.
Look for an image that has both those colors, right. So now we talk about segmentation.
So this makes me go off on a brand new direction matter segment.
Anything model. Some meta has a new machine learning model for things like that is called segment select.
This is just image learning that you're not doing um segment anything model.
Yeah this one. So please look at segment anything. It's pretty cool okay.
So at this one the when the neural network network has been trained to cut things out from anything else,
from a pretty busy scene like this can cut out just the horse. I mean, look amazing that is Hollywood has an entire, um, job classification.
It is called Mac pulling. Okay, so your photograph an actor or actors in a position, you pay people to cut them out using Photoshop like tools.
Okay, now those jobs are obsolete because they can cut them out.
No need for green screen. Green screen, meaning that you make them stand in front of a green wall and you can automatically do it.
But this is not a green wall, you see. So Sam is pretty cool.
You can cut people out of that. Wow, cut buildings are cut.
You know, cut the snail out and the frog out separately. You know, on the tortoise it knows like a lot.
Right. And then for people you can cut them out. You can know poses.
Well the sam is very neat. You can try Sam.
Sam is available by the way. And then also after they cut it out they give you point clouds and you can render the point clouds.
That is called Gaussian splatting. You can make your own render server. So it's pretty neat.
You can even cut things are some work. Okay, that's pretty amusing.
Okay, you can do demos, okay, you know, and upload your own dog and then see knows what the dog's boundaries in

your lives are.
Autistic, by the way. That's actually very surprising, right? The Eagle demo.
In the past Eagle would have said this all part of the Eagle, but then Sam wouldn't do that.
Okay, so let's go come like a long way. And then so, uh, Sam is like pretty neat.
Likewise, as you probably know, YOLO. So Yolo detector.
I'll play a little video for you. YOLO is also sorry YOLO detector.
This is also, uh, a segmentation model.
And so now you have YOLO version nine,
the latest incredible question what would you build a website so you can use all of this classification with more than 200 million users.
That's all. Six years ago.
You should look at something. Okay. So Yolo.
Uh, three years ago. Yolo v eight. Doing that.
Okay, so let's do you all over here, okay. I think it's up to version v9, but they should be enough here.
This is every new version. Brand new things come out a yolo V8.
Here you can see an example of basketball players and the jerseys, even possibly the numbers.
Example of how you can train your car. Amazing. That is with a custom data set to detect objects of interest.
By the way, it returns to Json with bounding boxes. The popular object detection model YOLO and it dropped yesterday.
You can try YOLO tonight. We'll go and kind of dive into the history of yellow and how Yolo V8 was made.
And then we'll look at a little bit about some initial evaluation. We'll go a little faster.
But there is a convolutional neural network. Talk about a bit about what you basically an amazingly cool CNN is out and what we recommend.
Maybe there should be a transformer version of YOLO.
Meanwhile, Vision Transformer YOLO version literature started working on shadowing.
That, as you know, is the classification probability system absolutely knows what is what replied to 100% uh,
where um, the PyTorch training was starting to replicate the Yolo v3 darknet training.
And then it became clear that, uh, the PyTorch training actually started to surpass that.
And so, uh, and so, uh, it kind of grew and we're going to keep going.
And uh, they were, you know, pushing Sota a little bit further.
Okay. So look at that. It's organized a knack where they're pulled together.
So please learn all this. Okay. For what it's worth, just for a head which is used to make the detections and the detections,
uh, these are all the different convolutional layers, like max pooling.
Um, concat has a, uh, batchnorm in class loss based on bounding box anchors, which is what, the other model.
It'll be kind of like, uh, the neutron, uh, pool. And then opening up the detection layers in a different visualization.
It is so robust, by the way. It looks pretty amazing when you look at that.
Uh, so in the chess chessboard, obviously pick up all the pieces from different orientations.
Okay. Uh, each time you go through each epoch and, uh, very clever and convenient technique to do.
Let's dive a little bit into the accuracy. Okay. So that's how like yellow, uh, comes in different sizes on how they do, um, relative to your,
um, and then how can that repository and then hopefully, um, what else should I tell you?
I think I told you about YOLO, Sam. Pretty nice. I have many thoughts in my head when I hear like, Google talk, like I was talking.
I think I told you almost all of it. Okay. So anyway, that segmentation right of segmentation and then kernel layout, you know,
I told you the old methods, you know, did not even look for a butterfly or anything, right.
Just simply look for colors. But the newer models know that you are looking for a butterfly,
that it can go and do things like YOLO in the database and pull out all the butterfly models.
So much, much more useful okay than all this, but we have to start somewhere.
Oh yeah. So color, texture, shape. Those are the three crude things that the older models look for.
Colors pretty obvious right? Texture means frequency. Very smooth wall like this wall or the blackboard.
Right. There's almost no color variation. Whereas my shirt has a little bit more patterns.
Right. And then the carpet as you learn more pattern it's called high frequency.

This is called low frequency. There's no detail. So you can look for a frequency detail that is called texture.
And finally you can look for shapes like logos and car shapes and things okay.
And so these are all ways to find similarities. And then you can even combine all the similarities.
How similar are they in color. And also added to how similarity in shape and how similar than texture.
Here the combined similarity is pretty high. Then you probably know that it's a good match and you can do it.
Okay, okay, so this is where the new cool thing starts, right? This notion of embedding.
Embedding is also a neural network that has to be trained with the same backpropagation to learn to embed properly.
So it is trained and trained and trained so that it learns to embed all the cats in one space,
all the dogs, all the ducks and all the guns in different spaces.
Okay, so once the embedding is trained and you have all these things that are properly embedded with the training, meaning you have many 100 billion images that are all probably in the right places in the abstract multidimensional space.
Then we can do an image search by simply uploading some brand new image,
a picture of somebody show that you liked on the street, or some dog you wanted to identify what breed it is.
Just take your camera, take a picture. Obviously the training data was not part of it.
It's not part of your training data, meaning it's a brand new image. And tell the system search for a dog.
Just like this dog that I just saw. Search for a product that I saw this other person use, just like, you know, in your database.
So no words, no keywords, nothing, right? It will pull up the very best matches possible.
So that is what this is. The image embedding is pretty great.
And image embedding follows. And now we call it multimodal uh large language models.
Meaning at first came text embedding. And then now we have image embedding.
We also have video embedding. Audio embedding. You can basically embed anything.
You can also embed chemical formula. You can embed uh bioinformatics DNA sequences.
You can embed circuit diagrams. You can embed architectural plans.
You can really embed anything at all machine designs actually for machine designs because embedding is just simply you find certain features,
meaning you find certain columns of data you want to describe. And those are what get embedded okay.
So there's no limit to what you can embed.
But you obviously have to come up with a proper embedding architecture to embed something new that, you know, a system doesn't know, but cool.
So that is all I care. You first trained to embed things properly.
So after the training is done, make a database which you call a vector database.
It's called a vector database of embeddings or simply column embeddings that have many billions,
if not hundreds of billions of things that you want people to search for.
There could be music. There could be videos, that could be images. It could be anything, really.
So that is your database, your query, your search query, information retrieval,
search query is now another piece of audio or an image or even a video, you know, or a piece of text for which you want similarity.
So then this new thing that you are trying to search for also gets embedded by the exact same embedding neural network,
meaning it will then go and hopefully land in a place that already knows about.
I want to get back here. Okay. So I have fun, right?
Otherwise, what I'm saying is, once you have an embedded neural network that knows how to embed things,
then you can use it to embed all the cat pictures. That would apparently probably go here.
Right? Or the dogs would go here and all the birds would go here, and then all the crocodiles would go here.
Then if you want a picture of something and you want to know what it is, take a picture.
And then you then search. That search will then go through the same embedding and do the search.
Then your query picture ends up here and say this all already cats because you told it that cats then suddenly know that it found a cat.
Other words, the same embedding architecture that was used to create those vectors is also being used to now query a brand new vector.
It's all things you have seen before. Okay, so again with TF, IDF is pretty similar.

The tf IDF embedding space already exists.
You make a new search on Google, a bunch of words, then those words will become a new query and it does similarity search.
So there's a whole similarity search over and over and over again. And then also once it finds similar vectors meaning similar images,
you can rank them in some kind of an order and hopefully give the very best results possible.
Okay. So I'm not going to spend too much time on ranking Reranking.
But overall idea is just simply, um, image embedding and search.
So look at this. If I just say image embedding, image embedding.
And I'm going to say uh uh, similarity search, then you will actually see an image that I can show you and that will illustrate everything for you.
Be this one. So this is actually, you know, I mean, you already know about Elasticsearch.
That's actually a thing you can download is written in Java, right.
So now you can use Elasticsearch memory similarity meaning that that's been like you know like here.
So dense vectors vector representation. So then that is what you have.
So all of this called dense representation because you already have a whole bunch of data that are embedded when your new query comes in.
Your new query is also a very similar thing, meaning it's also an image or a piece of audio.
That query is also embedded. That query then becomes central star.
And then you look around and you do nearest neighbor, meaning suppose this is the query that came in.
And these three circles are part of 100 billion. All circles are over here.
You are basically returning these three as the closest match. So very simple idea okay.
But it's an amazing idea. Works. So how do you generate embeddings.
You need like an embedding neural network okay cool.
And thus for image embedding by the way you can actually go to Huggingface and then use one of the embedding models already.
You don't have to write anything from scratch. None of this. You need to write from scratch.
That is even more cool. You just have to do it. Okay? That's all. Okay, show you one more picture.
Maybe pick some random. Mhm.
Yeah. So in the end users are going to, you know be in heaven because you can search for any roast type.
So you love flowers okay. And you come across a brand new roaster. Never seen it before.
It will tell you work in roasters or work in a cloud. What kind of mushroom.
What kind of fruit, what kind of dog? This endless but no words.
That is the brand new thing about this. Like, yeah, right.
Again, the encoder is a neural network that is going to take it has been trained to encode all of cats into one space, all of the dogs into some other space. So your search is also going to go to the same encoder.
And if you give it a dog picture, it will go to where the dogs are and it will retrieve only dogs.
Quadrant is a vector database written from scratch to do nothing but vector embeddings.
Security oriented quadrant. So quadrant is neat.
Look at this quadrant vector db.
I'll name a bunch of them, but we'll name quadrant for now. So cool.
Open source docker pull. It is that simple.
You can actually run it in Docker. So that way you don't have to download anything.
And then you instantly like start doing stuff, right? Yes, it is API.
I'll just say quadrant. What if you say quadrant vector DB similarity search okay.
I am making a homework for you and is fully not ready yet. I forgot something very important to you.
Like this has only one more homework. Three weeks right?
This week and two more weeks running for three weeks. So I won't give you two more homeworks and make you to work till the end of the last day.
I'll give you one more homework. So just leave it for homework. So should be fine.
Okay? But the last homework will be like. One of these things will definitely be.
And I am a drag. You know, vectorization. You will see. Okay.
So then you will get something when you say this, all you guys are same idea.

Then I'll just do one last one. Um, similar image search again.
They already use ResNet like a bunch of things. And then when you give it like a new vector is going to do a similarity search.
Actually I want to go here. Yeah, yeah. So you can find duplicates.
All the things we talked about in the previous part of the course can all be done in a more modern way.
You can also do a recommendation, obviously. Right. If you upload a picture of a donut it will then recommend you have pretty similar donuts.
So can you in what restaurant sell them. It'll be fun for people I think.
You know. And then like I said, the people that benefit from all of this end users because there's no SQL, there's no like what?
You know what to type these days in search engines, as you know, unless you know what to type, you're basically stuck.
So that's like a, you know, big thing for a lot of people. They cannot search. All right.
So this is all about reranking okay. But in the end it's all about embedding.
So Bing is using it. And Google will most certainly use it. Yeah.
So again same idea similarity. It's all about similarity.
Yeah. Happy parent pairs. Right. So tricycle is using it's looking in the image.
So not just the keywords anymore. So it's pretty neat. It's all about ranking.
So here comes ImageNet okay. And we'll take a break after this. So ImageNet came from WordNet.
You can see for example a cruiser squad car police car all some kind of automobile.
That's a vehicle. Vehicle is a transport device. It's called a conveyance device.
And that might be called a machine. And it might be called a human manufactured thing, what we call a solid object.
We can just go on up the hierarchy. Right. But then that is WordNet.
So ImageNet is basically a, you know, based on WordNet.
But now images face, you know, went through all of this. I want to repeat what she said right there.
So you can go all the way from very specific thing to a general thing.
And then now back to images again.
A car has so many parts you can label. And then the idea is how do you find that?
So like that just to open the crowdsourcing thing that she talked about,
if you did not know Mechanical Turk, Mechanical Turk is a thing that Amazon owns, okay.
Anybody in the world can go and then put a job out and a mechanical Turk and ask the people in the world bid for it.
In other words, I have a thousand documents to translate from English to Chinese. Tell me how much I want to get paid.
Everybody would propose like a number, right? And you pick the person that wants the cheapest money.
It's kind of sad, okay? Put the whole world against each other. Okay? So you will actually win anyway.
That is what it is. So that's what they used to label like all this pictures.
All right. And then. Yeah. Um, imagine that is, like, pretty accurate, you know?
Mechanical Turk. Cool. Wow. Look at this. Oh my God.
Huh? If you label 300 images, you get $0.02 to spend on the road.
Okay. Sad. In other countries, $0.02. Go a long way.
Multiply by 50. Multiply by 100. Right. Wow. 300 pictures.
$0.02 for 14 million pictures that spent under 1000 bucks.
Oh my God. Okay. All right. So let's get this.
And the rest is all just simply making sure that it works all right. You see that?
Obviously, if you give a picture of only one person and ask, what is this?
I'm pissed off that the okay, I'm going to say there's an apple or right lemon on reverse or purposely misclassify that right.
There is no what's happening. They send the same picture to a lot of people and then throw the outliers.
Okay. So that's how they can protect themselves from misclassification on purpose.
Like no one can screw it up. All right. So then that is all it is.
So it's so cool. Yes. So diversity in ImageNet applications.
You can do all these things right. So once the machine knows you can do recommendation systems, you can do like everything else you know.
Okay. Pros crowdsourcing cons also some kind of crowdsourcing worldwide exploitation you can say.
But you know by the way these people are researchers okay. They I not at Google.

So at the university we don't have lots of money. So we can justify using Mechanical Turk to, uh, crowdsource the labeling.
But Google shouldn't do it. You know, that actually goal is not ethical pay people okay.
All right. So data set ontology. There's a hierarchy. Amazing. Um.
Fey. Fey Lee imagined that.
Winning entry. Winning entry. 2017.
I'll read this and then a break happens. Okay.
See that she was at Urbana-Champaign and then they kept doing the same thing over and over.
Right. Small little neural networks that didn't go anywhere. And then she said, I'm going to do deep learning.
So first of all, make ImageNet. And then afterwards you won't go in it.
Okay. Great. And then ImageNet challenge, you know, like this by doing all this right.
Um, computer vision PR yeah, yeah, that's actually what happened.
So imagine that accuracy at one point became as high as 97.3%.
That is almost as good as human beings, right? Like wow okay. So then that is what we have to do.
And you can imagine that. So why don't we take a break from 646 to 656 and please come back in ten minutes, all right.
No more things. If anybody wants to come and sing or something, they can please, please, please come and sing and dance.
Which do you want to do in? Anybody.
But back at 656. Okay. Hey, I'm six, maybe a little bit far away.
Hey, check this out. Eyeglass repair kit.
Oh. All you need is a little screw, I think, in a little handle there.
We had glasses. Screws came off. Hey, check this out.
Check what out? Check. This out.
Uh oh. Uh. Hornsey.
Cool. Dance.
Oh. There was supposed to be the shock here and not this person.
Dash. Dash, I guess, is dash Patel. Push, push.
Verdun or no the nation.
Big trouble or. It can't be out.
Okay? Sharma. Where?
Okay. All the way in the back. Cool. Super.
Sugar. So we're.
Gonzalo. Here or not.
Oh, no. That's two in a row. Missing sewer and gangs out.
Unless they're in the overflow section. I'm not sure why.
Yeah. Croatia. Uh oh.
So, you know, it's a zero sum game. It means when they're absent, right, they lose points.
It's more for you. Your grades would go up. Okay, so grades would go down.
Wow. Uh, it's just for and for students.
It's. So many are on my list. Which one? Yeah.
Thank you for being there to break the chain. Okay. Graphic.
Quote. One more. In U.N.
In uni. Where? Okay. Yeah. Okay. So we can stop with anything.
Okay. See, now we can continue with a little parade of topics and things would go off on the side, right?
It's very useful. Uh.
Three blue, three blue. One round thanks to this here.
So it's actually said three blue one brown,
which is an amazingly wonderful bunch of YouTube videos that are explanatory that like, you know, backgrounds, right?
Apparently they have one on Transformers, a series on Transformers, which said, so let's look at it.
But I still want to save it for next time. Okay. Optimus Prime Transformers.
Transformers 11. Interesting.
Three B-1b. Yeah. You know, they should have a switch inside the channel, right?
It probably is in there, but I didn't see this.

Right. Cool cool cool. Yeah. And then when we come back, I don't want to get a bunch of people.
There's a woman called Julie Wang, Julie Wang, and there's glamor.
And definitely add this. Like, I come across many, many, many things and then some.
I think extremely high quality. They quickly get to the point. So I'll add this to that and we can watch some of them okay.
It is highly worth knowing, like I said, what the so-called transformer architecture is.
And then read this aptitude and see so that you have the architecture description on the one hand the code and see on the other hand and go,
ohmygod, I understand 100% of each line, then you totally get it.
By the way, Andrew is doing all of us a favor because there's many PyTorch implementations of Transformers,
but suddenly they make a weird call and you might not know where the college you cannot find it anymore.
So as you saw, this has no dependencies on anything. So 100% of the whole transformer architecture is in one C file.
That is as good as it ever gets. So definitely there's some advantage in doing it this way.
But there is some news for you, for all of us.
And this is less than two weeks old, which is the whole architecture, right?
Large language model architecture is based on this notion of this thing called a transformer.
Transformer. And the transformer is an architecture with all the stacked encoders, you know that I showed you.
But underneath it all, this is very important idea called attention.
Attention is a matrix, like a square matrix with all of for every word.
The words go this way, words go this way. What word has to pay attention to what previous word is?
Attention idea. Right. But very, very interestingly.
In other words, attention is at the core of all of this.
So now we have multimodal variations. We have long chain.
We will learn all the things. Right. But attention is at the bottom, the bedrock of everything.
And that is the paper that arbitrarily people draw. Attention is all you need.
But incredibly, it's a brand new transformer that is not based on attention at all.
Like what? Oh my God. So the one thing that we thought is absolutely necessary is not necessary anymore.
And they claim lots of advantages. The new architecture, because attention is quadratic, you can only pay attention to so many words.
That is called context limit. Let's read it okay 60 4k. Beyond that it becomes so impossible that you cannot keep up okay.
But the new architecture is sub quadratic. So it means you can pay attention to longer and longer words.
What the [INAUDIBLE] is in your architecture called mamba? Oh lm lm mamba Mamba bang!
Bunga mama mama mama.
Tactics. Huh? Oh, cool. So man power is incredibly crazy cool.
It is a new paradigm. Select. You get the idea. Get the job.
Transformer. Get it, get it. Transformer. So it is most certainly pretty cool.
And most interestingly, Mumba goes back to an older type of neural network called a recurrent neural network RNN.
Attention came along and made fun of warnings and said, RNNs, LSTMs, they all suck in our attention is a new way to go,
but Mamba is actually based on an RNN like architecture, and so that is actually where they get their sub quadratic, you know, attention computation.
It's not even attention. They call it look at it okay. It is called a structured state space sequence S4 model.
There's no attention computation anywhere. So please pay some attention to actually do it to understand how this works.
And I say look, it is still doing what looks like attention, but they'll never call it attention okay.
This notation. All right. See? Like this.
Simplification. Lighter, faster scales linearly with the length of the sequence.
That is ridiculous. In other words, the one from an open square model, which is this attention model to an open model.
Whoa whoa whoa. Okay. Yeah. So neat.
Afraid not achieved by any of its predecessors. So true. Okay.
And it's also very simple because attention is like lots and lots of blocks and multi-head.
You know, it's a pretty complicated thing, right? This one is looks pretty simple and even more amazing.
Those are recurrent things. The units can actually be hardware accelerated.
Even better. So yeah. When we say hardware, it's overrated.

Like. Like. There's nothing like.
It's like. It's like.
Of hardware you're describing like some.
Yeah. I mean, it's basically simply design, you know, it's an attention computation.
Yeah, it's quadratic attention. But they're what's called multi-head attention,
meaning it's not just one block of code running through all the n squared that can actually compute attention in parallel.
Okay. That is actually what multi head attention is. So when you have a GPU with many blocks with many course you can accelerate that to some extent.
So likewise this linear space calculation can also be accelerated.
So it's an yeah exactly. It's mostly same this GPU course.
And ultimately that that's all it is. Yeah it's very simple actually. Yeah. Like that right.
Cool. Yeah. So maximize parallel processing I mean it's going to be pretty neat.
So the compare are basically mambo transfer. And please read okay. It's like very neat.
And I tell you how Transformers work right in here okay.
So attention keys values you know okay. Attention. The decoder creates input layers.
You know self-attention okay. But it is quadratic though.
And uh okay. Length sequences attention process some number.
It's a different approach selective state space. So yeah I mean look at this one.
See that. N squared versus an inference on versus constant batter and everywhere.
Okay. So we should pay attention to. And it's so new.
You can go to Mumbai. You can even compare Mumbai with some existing transformer and see that it's actually fast.
Okay? That's AI, which uses Transformers.
But pretty soon that will be done by mama. Now I think it's going to be great.
Okay. Really. So I'm going to of.
Anyway, that's what I mean by saying practically every single time I look for things, I come across something brand new.
It's just one. But this is really, truly mind blowing because attention was the bedrock of everything game.
But now it's no longer true. So who knows what else is lurking out there.
Okay, so I want to get back, right? And then tell you.
This is what I mean by assorted topics, just so you know, a tiny bit of it.
Almost all of it. Bleeding edge. How do you make it up?
So where were we? We were in a slide here called search.
All right. So what if you make a search engine that is optimized just to purely retrieve code?
So now this might be called task specific language models.
You know, because code is very structured obviously in terms of the keywords.
Right. A small number of English words going into Java C plus plus, you know Ruby, Python.
So why not tune your search engine? Why not construct a search engine specifically for doing code search?
GitHub does exactly that. So when you go on github.com and start to actually search for words, then how does it work?
That is what this blog. So they have a blog technology behind code search.
So the bottom line is that in order basically that that training data is all just code.
So then that makes it optimized. So see here rust.
Oh I can go off on that tangent also.
But I won't go to my channel tangent.
But C plus plus is also questioned by the rust community because C plus plus is also the gold bedrock of everything.
There's nothing lower in the world, meaning there's nothing smarter, better in the world than C,
C plus plus on which the entire universe is built, including Python, including JavaScript, including Java.
Practically every language we use in the world today, even Swift, by the way, Golang, pickle language, dart they all look like C plus.
Plus they're the same for look. There's a similar statement, right? Rust also looks superficially like, say, C plus plus.
But rust manages memory better, meaning less crashes less, you know, looking at dangling pointers, you know, stuff like that, right?
So it's a more safe memory usage mechanism. Yeah.
So it's Java, but Java pays the price for amazing memory management, meaning safe programing by being goddamn slow.

Yeah, it can be ten x slower than C plus version gives a rat's ass, right?
It runs in the server, but rust manages to be as fast as C plus plus and still does a better job of managing memory.
That is why it's just amazing. Okay. So if rust was also slow, no one would care.
Like why you wouldn't have a new language, right? Uh, but rust is not that rust us other advantages going for it as well.
One is called WebAssembly, which I'm going to tell you. Okay.
So considering C plus plus the rest, you know, I mean obviously answers both the answers go both way.
But say this from Mozilla Foundation, right. The people the right Firefox okay.
So obviously they know what they're doing. And so you can read like much much more.
But at least most certainly, you know, be out there. Okay. Uh, see like this.
More features a few bugs, but the video game programmer would tell you, yeah, but you still cannot touch C plus.
Plus for bare metal. By the way, this one is the C code going to bare metal okay.
So in this case talk about training. You want the training to go as fast as possible.
And I didn't see what I didn't notice. You did not make this in Python.
I could have made the whole thing in that part. Right. But we're making a dot C.
It is actually much faster. So anyway, that's what rust is.
So then why do you care about rust? Are basically, uh, GitHub people chose rust to run to make the search engine.
So obviously if they didn't go as fast enough, they're not going to do it right.
And they have a past. There are many search engines. GitHub is like, you know, 20 years old.
So you have a history of search engines for GitHub and a little link there. But I want to get thrust here and tell you something crazy cool.
Rust is a compiler and its compiler is a WebAssembly compiler.
So the WebAssembly compiler will take the rust code, say Search engine one, and turn that into a binary called dart.
Awesome. This is a lot like dart class, where you go from a Java file to a dart.
You know, class file is very similar.
But this binary, this Wasm binary, it's a virtual microprocessor, you know that, uh, all the other Chrome uses, for example.
Okay. So this is how browsers work. But now suddenly anything that is not browser based at all video games, image processing, servers,
machine learning, all of it in the world can be written in rust and many other languages and compiled into Wasm.
So awesome has become a universal binary.
So in Wasm so function call, like any other function call, it has fixed data type in 32 color and then it will give you a function output.
So no matter what programing language you programing, they can all become Wasm binaries which can be mixed and matched.
Suddenly you have a new world. You can run Python based machine learning that you already have.
Turn them into Wasm. Right. New machine learning.
You know, cluster functions, for example K-means clustering in rust.
Convert them also to Wasm, but make this Python wasm function call the rest wasm function, because at version there is no difference at all.
Wasabi or wabi washi?
Throw it any acronym okay. So washi actually is a brand new thing.
Washi from Yarn. What the [INAUDIBLE] is all this?
What is what are you talking about? Wasm washi waggy.
What the [INAUDIBLE] does you have? Seriously, this should be in all the features, okay?
Because that is where software development is going.
Microservices can all be packaged in the little Wasm put in Docker containers run in some GPU cloud.
God knows where. All you need to know is the name of the function, what inputs it needs, what outputs it's going to give you.
You can learn mixed language programing as a programmer, so not limited to Python libraries JavaScript at all.
It is great. So the interface the application interface is called OAC.
So again some of the time we can you can go look at all this yourself. But it is just absolutely great.
See that portable size and load time efficient format compilation target for the web.
Wow. Just so many neat things. So what languages can you program?

Uh. Source languages. Source languages for Wasm.
In other words, what can become Wasm binaries? C c plus plus can become Wasm binaries.
In fact, I was one of the first compilers ever written is called Emscripten and scripting to Emscripten.
Suddenly C can become. Watson.
Well. Okay. You see here, right? Yeah.
Just simply load the washing model. How do you load one of these washing things?
One line of JavaScript. It means from your browser, you can run very powerful things.
But. But they're not running on the browser.
You can run video games if you want, but they'll run pretty fast because your browser is not running them natively.
They are being run to fetch. Use the good Ole JavaScript fetch command for doing the washroom calls.
Okay? And you can run Wasm outside as well. You can make standalone applications just like node or something, right?
Yeah. Require. Anyway it's just great right?
See this audio video 3D objects.
We not time to click on each one of them, but to prove the point. What I said to you, all these people have done all these crazy things already.
See that? Imagine all that being rendered using Wasm.
I mean, it's a great racing video game. Engines. Okay.
Uh. Yeah. But I still never showed you the languages.
Check it out. Look at this. CC plus plus first go.
[INAUDIBLE] is a programing language. By talking content.
Now even LabVIEW is like a machine that divides programing language for like, you know, instruments and things.
WebAssembly and then yeah, so many languages C like this just go on and on and on and on.
Like in progress. Pretty much every single language in the world with all the Com sources for Wasm binaries.
You just wait. And we can do mixed language programing.
Then there is no porting anything to anything else. But I want to get back here though.
Yes. So all of this said code search couldn't. Okay.
Yeah. So the easiest search engine in the world is grep.
Okay. The Unix grub command. Just a command line such. But obviously most of the world doesn't even know what grep.
And I'm going to do it okay. But grep is pretty faster. So this is very cool.
This is tfidf things you already know. Okay.
So you can make a list of terms that, you know basically need to be indexed.
Whereas like for a while return function in double you know programing language keywords.
Sure. Doc ID that could be a piece of search code.
And then the content is in this case some kind of definition call.
This puts. Okay. So in different programing languages you know uh doc IDs and certain keywords they're looking for.
See look N-gram. So words that occur together right. It is things you already know but then optimized just for programing.
45 million public repositories, including yours and mine.
When you put something on GitHub and you agree to the little checkbox that says, my repository is private, anybody in the world can browse it.
You give you a call away. That is why, you know they can take it. But in this case, they built a search engine for you.
Microsoft bought GitHub and said, we'll take all your 45 million repositories and train GitHub Copilot, you know.
So now Microsoft vs code Copilot and put programmers out of a job.
You know, that's pretty bad. Uh. Um.
Hmhm. Okay. So look at this GitHub class action class action lawsuit.
Say programmer in Los Angeles of all places.
Who's also a lawyer said screw this. You know Microsoft cannot do this to us.
So I made this massive case. You can look at it okay.
See this battle against intellectual property violations from Microsoft who took GitHub?
All the code that we wrote and made this thing called copilot, which will sell it to us for money.
This copilot was bought by Microsoft in 2022.

Back then, all of us wondered in the world, why does Microsoft care about open source?
Right? They're antithetical to each other, you know, the question of each other.
But now we know why. Because they took all the code that we wrote and trained the error, and that's actually what happened.
See, this subscription service is free for students.
But obviously when you make money by writing code, they'll actually sell it to you.
So it's a crazy lawsuit that's actually going to go on, right?
Yeah. All right. So if you wrote code, it means you might get paid for it.
Look how cool this is. It's a classic search. Indexing, right?
And then now there's this thing called blackboard. You can read the details afterwards.
Their their engine is called blackboard. Right. For the search engine.
But then look ingest crawlers again just crawlers means it goes and gets all the code that they need to index it.
Exactly like how you how you did for your URLs. But these are not URLs.
They're already GitHub.com stories okay. But still I have to go and collect them.
To go grab actual source code and then use things like Kafka and actually send it to all these shards, I guess.
Okay. Yeah, I didn't really like all the details fully explain to you, but.
It's a custom search engine that uses exactly all the usual search engine steps, such as crawling and doing the Tf-Idf reconstruction.
But all, uh, specifically for keywords that we can search for.
All right. Uh. Okay.
So I'm not telling you too much is not too much here. But it's still useful because where does it all actually, you know, play out?
You'll go to github.com and it type something you can type.
Well, what do you say? I figure, hmm? What?
Oh, this one, the carpet. Uh, sorry. Go up here. Uh, so sign in.
Yeah. Sign up. What the hackers get approached.
Yeah. Okay. So we can now click on say that I say the word fractal okay I'll say the word fractal.
I don't know JavaScript or something. That is where our search engine kicks in.
And instantly for 89 repositories. That's pretty neat.
Fractal jazz and jazz whatever. Right. They all have factor and they have jazz, you can see.
So this is the beauty of this. Okay. So we can then learn so much about coding in any language at all.
So you want to learn Swift programing you know for the US class across.
The room from here. Then you can go back and just do one last time, okay? It's pretty neat.
You can just say Swift. Just learn the type. The name of a language you want.
Beautiful. Three 10,000.
Oh, okay. All kinds of swift things. So that is pretty neat, right?
That was a search engine. Oh. All right, so that is cool.
Location based search. Proximity search. Cool. I'm going to tell you because it's such a such a search.
But a new kind of a more useful kind of search, in a way, is your phone always knows where you are.
GPS coordinates. Right. So when you search using the map software on your phone.
Nearest gas station. Nearest hospital and Indian restaurant.
I basically they have to give me things in the area, right? That is called location based search.
And that is extremely useful.
As you know, what is the point of sending me to an Indian restaurant that is 50 miles away when there is one just less than one mile away, right?
So they have to then take your location into account and take the location of what they're searching for, and then merge them.
That's quite useful, right? That is called labs. Proximity means near nearest.
So same thing in or near to you. Search. Quite useful.
Cool. So then this one is a nice article that tells you, look what that actually means.
She uses a smartphone's GPS. Right. That's pretty cool. And then now a real time location tracking.
It's all real time because if you ask for the same Indian restaurant ten minutes from now,
they have to suggest me something else because maybe something else is closer, right?

So you are constantly updating your search radius and your search look what falls within the radius.
Okay, cool. So how does it work? You know, like all this Wi-Fi?
Yes. The Wi-Fi can also track. In other words, you know, your laptop actually might do that.
It might guess where you are based on the access point that you're connecting to, for instance.
Okay. If nothing else, I will roughly tell you Los Angeles for anything that's in LA, because all of our Wi-Fi,
you know, the characters are all connecting to some points, like in L.A. somewhere, obviously.
Or you can use cell tracking. The cell phone towers can tell you what tower is handling.
You call the tower knows where it is then. Therefore it knows like you know where you are.
Because you know, by the way, cellphone towers are also Voronoi polygons.
So there's one cell phone tower, second tower, sort of a term.
Then each tower has a convex polygon that basically sounds where the towers region is.
So here's one tower, second tower join them. Perpendicular bisector.
Exact same as what I showed you with clustering.
So if you're driving along like this right when you're driving here, as long as you're within this polygon, this tower handles your call.
As soon as you leave and go away, it will hand your call over to the next polygon.
That might then pick up your call. So it's pretty neat that they're all arranged like that.
And because they know their own location, any any cell that is handling your call would know that you are within this region.
So it can guess where you are. Okay. Cool. So then you can track like people in all kinds of ways.
You can even track people based on like an RFID that they give you. And you can wear the ID or something and then actually pass by doors.
Machine can track you with your permission, obviously. So then that is all about alibis, okay.
You can use like, all kinds of neat things. Store locators, you know. Okay.
This is actually very cool when you're walking by a clothing rack.
It has some, you know, denim jackets that you probably have bought in the past. Suddenly, a coupon flashes on your phone only for you.
20% off right now. If you pick it up, there is hyper targeted marketing and they can do that with location based search, right?
So there's many neat things like even all this travel information.
Yeah. This is extremely useful right.
You know, for example, Mercedes Benz, you know, you an accident or something then that serious whatever they call that the system,
the navigation system will tell some Mercedes Benz operator exactly where you're stuck.
You're too short, little freaked out and anxious to even make a call.
And then describe like what your your incoherent call cell assistance to actually help you.
How, you know, location tracking. Cool.
I say this a lot, right? Okay. We are.
This is quite useful, right? I mean, this happens to us all the time. Then I am in LA, and suddenly if my card is used in Sweden.
The first thing that happens is the card transactions decline. They don't think I travel.
So you have to me to call them and say, now I'm actually traveling.
And all of these companies like Uber, Lyft, Grubhub, DoorDash, okay, all of them location based, obviously between the restaurant and the customer.
Right. So it is very neat. Pokemon Go was like an augmented reality application, location based, cell based.
So I need I'll tell you about Uber's hexagon system pretty soon.
But then not yet. Okay.
So yeah, these kinds of papers are very nice for you to read if you like.
If you like to talk about all this location based nearest neighbor search.
But these are things you might have seen before. You know, all of this makes sense to you.
Okay. So spatial network queries this a lot about spatial data but mainly what it is all of location based data.
Assuming there's a map somewhere and then all locations on the map. So then on that note Uber Uber.
S3 indexing. So what happened was in the world of spatial databases for many, many years, decades, there is one

indexing scheme called a region three.
It's called the R3, where you think that it's a big rectangle for all of North America,
and inside that there are smaller rectangles like a hierarchy for every state.
California is one of them. And within California, there are more rectangles, one for each city, maybe one valley.
And within that there are even more rectangles, one for different parts of the city.
And finally, there's one that surrounds the USC campus. It's a hierarchy of rectangles.
It's called the region three. It's called the R3. So R3, R3 spatial indexing.
This was a state of the art for so many years ago until over said there's something better we can do.
See, this is what art is. So say these regions one index.
You make a tree literally that says first level, second level, third level,
even smaller and smaller until you can get every individual building you want to index.
Okay. Okay. So there's nothing wrong with it.
There's been working for like many decades, but it is highly inefficient because the rectangles overlap each other and waste space.
And sometimes rectangles don't cover all space. Got gaps. Okay.
What if you tile the world into small little hexagons and those kinds of hierarchical?
Each hexagon can contain even smaller hexagons.
That is over H3. This is so cool, right?
H3 jokes are so Google invented this. So Uber invents this entire system and that can be useful as well.
So when you book an override that is actually what happens.
They partition the whole world, including where you are, what the drivers are into all this hierarchy of hexagons.
So you might be standing here, they'll match with all the drivers within the same lexicon, because they are most likely to pick up a ride, right?
Not somebody halfway across the city. It's damn smart. But that is, they own your system and you can try this as well.
It's actually on GitHub. Okay. When it is called select an H3.
So please give h3. I try to hack okay.
Yeah. Okay. So it's very neat. You can try.
See that? That is a modern way of doing location based search.
Cool. So let. What else?
Um, yeah. So this one is actually a little freaky.
It's probably not what you intended. Labs can be used to pinpoint your location and actually target you and hurt you.
Kidnap you, kill you. Whatever. Right. That's actually very scary.
That's possible because it is not supposed to be like, it's not supposed to be that accurate, but they can actually track you to where you are.
But it's been done. Uh oh. Discovery attacks location based pregnancy.
The sad thing about the world is there's always some bad guy or bad girl trying to exploit every gosh darn thing in the world.
Leave it alone, okay? As soon as something good happens, how to attack it?
But that's what cybersecurity is all about, right? So that is something you have to read and how they attack.
Yeah. You attack basically by doing correlation between many different data points okay.
In other words, one location information by itself is not precise.
But when you combine multiple of them, you are able to actually, you know,
basically triangulate and get like where somebody is within a certain view in 500ft.
If somebody wants to kidnap you, all they have to know is like, what block you live in that can hang out for days and like, you know what?
You come and go, okay. Yeah. It's not that hard. So that's why scary stuff like that.
So you need to then make sure that these algorithms are also a little bit safe for like end users.
End users couldn't care less about any of this, right? You invented it, but you also made this loophole.
That bad guy comes and exploits. Not good. Okay, let's talk about vector database a little bit more.
I told you bits and pieces you earned from starting from day one. Okay.
So maybe there's not too many new things to say, but there's always new things to say.
Pine cone is just simply one of them. Wait.
Time to list a bunch of them. Okay. Pine cone is a company in San Francisco.

Not that far from OpenAI. So pinecone wrote code from scratch, and Sarah will become the leader in this whole vector database world.
Except the concept of vectorizing and embedding is not new at all.
It's at least ten years old in some of the slides.
Okay, but because of the new algorithms similarity search, it became a new need, a suddenly brand new need.
So pinecone is out there. Pinecone has a library called canopy.
Canopy and canopy has a very special purpose.
It is for doing what is called retrieval. Augmentation. So retrieval augmentation is a very easy idea also, which is you have an algorithm here.
And the user it's an alarm. User queries tell alarm a sentence as a video query.
And then the query is going to come back with an answer. Right. But ChatGPT is not an expert in every single topic in the world.
It might not be an expert in JavaScript, asynchronous. You know, programing in the code works.
It might not be an expert in nuclear materials, okay? It might not be an expert in arms reduction in our ending war.
Obviously it's going to lie, meaning it's going to make up crap. Right?
So the obvious thing to do and the wonderful, amazing good thing to do is tell the lamb to use testing that is called external memory.
External memory, even though I call it external memory, it can be actual memory meaning like a database,
you know, or these days it's been more generalized to running any code you want.
It can be Mathematica here. We still call that external memory. Okay, so that's a pretty loose term that we use.
But the idea still is the same,
which is don't let the see and query instead make take the query and also do a search in this external memory one way or the other.
And the search will return some search results. Right. Those results can become what is called context, meaning your prompt.
Your initial prompt can be expanded and augmented with actual good answers that your question is looking for, but make the rest part of the question.
And so now your prompt has become much bigger. How did the augmentation happen?
Because you went to the external memory? Then that is what Al-Ahram takes and spits all that extra back to you in natural language.
That is super cool. So we completely bypassed all alarm except to understand what the query is about.
Okay, but it is not answering it.
So we call that retrieval augmentation retrieval retrieval augmentation, retrieve mean retrieve the answer but not similar LM.
It's augmented by this external memory. So Canopy Library say this external memory was a pine cone vector database.
Pine converted database. What does it mean to say take a query, you know, do a search and come back.
All right. That is all done at canopy. But just a few lines of canopy.
You make this whole rank process almost like trivial ten lines of code.
Neat. So that is pine cone and canopy. Likewise, here are some of the other vector databases.
It is worth running something in each one of them. There's not too many of them.
And then your resume here will just shine. Chroma. Chroma.
One more call Melvin's. Believes.
These are some of the top ones, right? Okay. So these are all custom built from scratch just for doing vector databases.
But what about databases already around like Oracle.
It's the oldest relational database in the world, right? What if they post add?
What if they retrofit what they have? In fact, even store vectors as relational tables, right?
Doesn't matter. But they have vector support, meaning you can do vector stuff at Oracle.
So that is becoming possible. So Oracle has vector support.
So it has Postgres. It's called PG vector Postgres kind of vectors.
Readers can also do vectors where this is a main memory cache in-memory database.
You know that can be run in like Amazon. So they have vector support.
So there's a few others as well. So relational databases also have vector support.
So I'm going to Google some of them just to show you what I mean okay. That's worth spending time on.

I'll just say Oracle Vector Database.
See. That's interesting. Right? So less than a year ago. That is pretty neat.
So suddenly the oldest relational database, the oldest database company in the world.
Period. Do something brand new. That's 2024.
Wow. Right. So let's see what they say. Like this.
Semantic content of documents, images, unstructured data as vectors.
That means as vector embeddings. It's like all these things that I do over and over, you know, and then your query also becomes a vector.
The query is not SQL query. It is also not keyword query.
A music query is your hum a piece of music, your image queries.
You take a picture, a video query, your video something. And so this is so cool right.
So you can just run. Yeah. Similarity. So we call it rag right a breakthrough technical call.
This rag is the coolest thing.
Any company in the world will not pay you guys money and hire as programmers to write some fun little application where somebody says,
make me a panda smoking a cigar. Let's all like fun and cute, right?
But it does not, you know, pay the bills. Okay? But every company wants a good chat, but it can interact with the customer as well as a human being.
What? No chat bot like that, as you know, exists in the world today.
They all suck. They're all cute. They're all script keyword based.
As soon as you go off script and say, hang on while I get something to talk to you, that's a waste, right?
But with this, finally, the idea is that rack system can memorize, so to speak,
a whole product catalog about chemicals, about drugs, about medicine, you know,
can help doctors go through all cases and summarize it for them,
can go to lawyers and help them go back ten years and summarize lot cases that happen, okay, there's no end.
How many cool things we can do with this. So rag is the coolest idea in my opinion.
Because it is. It provides high accuracy and the training data.
Meaning the core LM itself was trained using openly available stuff, right?
Your company stuff is obviously not part of it.
Okay, so by using rag you have no worry, no danger that somehow you need to basically, you know, retrain the lamb with your data.
I guess what I'm saying is you can run on a cloud, you know, little alarms, you can run a cloud.
So the opposite of doing rag or something called fine tuning.
So fine tuning means leave the rag idea alone, but take the lamb and then add extra data.
A lamb already has so much data. Add your company's data and make the transformers themselves know about your data.
That is called fine tuning. But if you do that, then your train transformer is maybe sitting in AWS.
Somebody could steal it. Okay, so all that is not the case when you do rag.
When you do rag, you can keep that external database right in your own servers and absolutely make sure nobody else gets it.
Okay. So it is pretty safe when it comes to trusting, you know, people with data.
Okay, great. So that is all. Keep on going. I want to tell you more though.
What about, uh, Postgres vector? Database vector.
PG. Yep.
So let's see what they have to say. Postgres says something very similar.
Compile an extension of this. Right. Great extension writer.
Okay, this one is actually telling you how to use SQL. See this is amazing.
You can do vector search using SQL. Wow. Good old SQL syntax.
That's pretty neat. But I want to show you though. What is a vector?
Yeah, exactly. So like here. Okay. So vectors together data operators.
So what they did was extend SQL in a way to make it vector compatible.
That's pretty brilliant. So you can learn you already know SQL but know the vector stuff you know with SQL okay.
Then they wrote this article. So much for you to know.

You know, if you just know, like, where to look. See that right there? So in the end it all comes down to similarity.
So you see like right there you can do cosine similarity in everything you've seen in this class.
Like right there. You can now do it with uh Postgres.
What about Redis. Yeah, a type or a then comes up very neat.
So then you can go to a registered URL and actually. Oh and actually set up uh products key features.
I'll just search here okay. Can we search. If not go to a different link okay.
No one. Yeah. So let's see what this says.
See vector field similarity search you know and then index. Yeah I'll tell you about the indexing pretty soon.
Indexing is pretty amazing vector that can look K-nearest neighbors right back to doing nearest neighbor search that one.
So when you when this is your query there are so many other documents.
K nearest neighbors is literally the k closest existing vectors that match your query.
And then it'll rank them somehow and then start to present. Okay. Tarquinius neighbors.
And then this classic Euclidean distance right there you can say, huh?
Oh, I love this. Create a vector field like in physics. Okay. Kind of a vector like kernel field, but this one is not a vector field.
Yes, brute force algorithm is actually very slow and very painful when you have lots and lots and lots of vectors.
What do you mean lots and lots of vectors? 100 million vectors.
Or maybe 110 billion vectors. You cannot do brute force.
Brute force literally means take your query and find a distance to every other vector that there is not in two dimensional space.
By the way, you maybe 10,000 dimensional space squared is going to be squared plus squared plus squared plus squared plus or 10,000 times.
And take the square root. It's crazy right. It's not going to scale.
But instead we can actually index the space. So we'll get to that okay. That's what flat versus meaning.
If you said in some kind of tutorial that you do to flat versus you will see the difference like R3.
That is almost the same as ordering region three but for multi dimensions.
So all you have to do is just type what they tell you okay. You literally tell, you say import numpy as NP everything is python.
You already know what to do on a PC. Please install Anaconda.
Python. Anaconda. Python distribution.
You know, you might have heard this already for your homework. On a mac, you can simply use a mac terminal.
Okay. So that way both of both parties, both users will have some kind of a text terminal to type all this in.
Yeah. Very neat. Like right here you can really set vectors.
You can take a sentence and vectorize it and say show me what the vector version of the sentence look like.
None of this is magical. And then you can even find a right type of similar sentence and say do a course in similarity.
Tell me what the cost dot product is.
It will be a pretty high number and make your sentence be very different and say make me a cost dot product, show it to me.
Then they will almost drop to zero so you will know exactly how this thing is working.
Okay. Lazy. Mental. Transparent.
Okay, so K-nearest neighbors searching us. All of these. You can try.
But now we are talking about right is right. Very neat. So then I'm going to give you a tour of all of them.
I'm going to say quadrant quarter and vector database. And when it comes to pinecone I'm going to tell you about pine cones canopy okay.
Great. So docker pull docker run to make it so easy for quadrant.
What about Melvin's. Mel was Mel was not.
Yes. Sorry Mel was. And Mel the US mill was.
Mel was is one of the first actually. Okay. So you know again you see that there's like vector search going on.
Watch the video. You know we should do it anytime again for oh hold up, someone just got Bogo free Blizzard treats.
Oh in the TCU app. Sorry. So I can imagine right.
Each one of those might take many hours. I would rather tell you the bread than just going to one of them.
Thank you for very good. Welcome to an intro video on Miller's, The World's most advanced vector database.

Our world is full of data. In the early days of the internet.
Data was mostly structured and could easily be stored in big tables called the relational needs SQL research.
But as the internet grew and evolved, unstructured data became more and more common.
Emails, photos, protein structures, the list goes on and on.
As the industry evolved, the need for a way to understand all of the unstructured data grew as well.
Machine learning, specifically deep learning, fulfilled this need.
Deep learning models can, generally speaking, be used to help computers understand human generated data.
A computer's representation of this unstructured data, called embeddings, are essentially high dimensional vectors.
If two embeddings are very similar, it means that the original set that is ultimately all that it comes down.
This isn't that slightly different, obviously, to different animals like cheetah versus cat.
But then the pose is very similar in all very similar. Right. So they end up in way.
Okay. So I draw two dimensions. Imagine a third dimension four dimension five six 10,000 dimensions.
So all those numbers are dimension is simply I mean, the 10th dimension is simply a list of Python lists where each number is one value for each axis.
Okay. So then all those numbers will become one point.
But it's not a 2D point. It might be a 10,000 dimension point, but in the 10,000 dimensional space,
those two points will be damn close to each other because they're all based.
If this was also zero, right? The distance between them is literally zero.
That is the closest match you can ever find, but set off by one number.
Then there is a tiny gap between them that is still the closest distance ever for me to be arbitrary.
Okay, so it's very easy idea. Where do we store these embeddings?
How do we maintain them? And more importantly, can we make it easier to do lookups and other tasks for the searching the right index?
Everything moves as an elephant, I did a foundation open source project for the singular goal store index and manage massive quantities of index,
unlike existing relational databases. Okay, look at that vectors.
As second class citizens, Novus is designed from the bottom up to him.
That is true here. The second class is starting up again.
In those instances, using Docker is impossible.
Simply grab the latest docker compose file from that simple start up the image file and begin using numbers.
We provide Python and should all try this out for easy integration into your new JavaScript of Python.
From implementing a question and answering system in a variety of languages,
to searching for products that might be of interest to us, that is for Amazon, right?
Shine Alibaba what? They don't love it.
Apple research payments limitless chemists material centers 2.0 movies has moved from being a
single instance package to Music Vault Shazam scalable to trillions of embeddings and beyond.
You know, how do you do trillion embeddings, right? They're not just saying that you have to index that space.
You know, this in this space right here.
How do you index especially what I can tell you right now, supposing you got all these points right, like all over the place, and then you got some kind of query that comes in, you want to quickly find the closest neighbors, right?
One of the easiest things I can do is make a big square, a square, a little square in 2D in multiple dimensions.
Same idea that bounds everything. So I have a square.
Then I can subdivide the square into my second level like recursion.
So my first level of hierarchy is one square in which all the points lie.
Then I can have four way subdivision smaller squares on which the first bunch of uh, embeddings lie here.
Second embeddings lie here. Third one go here for the lower here, but I can keep going down.
Go and subdivide. Right. Go subdivide I can subdivide, subdivide, subdivide till every single point actually goes in one leaf node if I want.
Okay, that is called a region tree. I can actually just call a quad.
Transitional region tree is called a quad tree quadtree quad because it is always quad for division in three dimensions like Minecraft or Roblox,
or tree because it is in a cube. You divide eight little sub cubes, right?
Because you come back, come in the Z dimension. So now watch how amazing this is.

Suppose you want to do a query here, right? You wonder what should I pay attention to?
Look at this. You take the query coordinates say this x y.
You know the query is coordinate, right? Meaning the vector. And go here and say in which of the squares is my query point?
Say my query point was for some reason here, right? It means you throw away all of this.
That means you don't have to look at these at all.
You throw away 75% of your data when you have 100 billion vectors, suddenly you have to only look at 25 billion vectors.
So why stop there? Right? That is divided into four parts.
Remember my query was here, so then I can go in here and say all of those.
So now I know I'm in that quadrant in which further sub quadrant should I go?
I go in one more sub quadrant which is that sub quadrant. So guess what?
I don't have to search through all of these at all. So like doing binary search.
But even better in binary search only discard 50% of the data here.
Discard 75% of the data looks so damn efficient.
So in just a few search. You're pretty close to where you are. Imagine doing this in 10,000 dimensions.
Okay. Same idea. Whoa whoa whoa. That is called hierarchical.
Navigable. Small world. Hierarchical navigable.
Small world. So when they say index vector data, that is exactly what they're all talking about.
This algorithm called hierarchical hierarchy navigable because you can trust them.
Small world. You know, each little piece is called a small world okay.
And get really smaller. So that is the indexing algorithm. This algorithm entirely saves the day.
So pinecone io has, like, all kinds of cool things, right? See this?
Exactly what I told you. Okay, I had a pretty deep level.
You know, you would go down this path. That means you don't have to go on all these parts at all.
There are the next level. Go down one more path, but ignore all of this and all this.
So in just a few steps, you're able to very quickly get.
In fact, I'm going to show it to you so rapidly, get to where you want to be.
You're going to be having a look at the hierarchy unnavigable small worlds graph or NSW graph.
If I should speak like a magnet, it's a small world. Technical, magical, small world.
Okay. How it's used in vector search.
Now, patience is what you said.
It's actually search algorithms for.
Approximate nearest neighbor search. So fast forward in a minute.
What I want to do with this video is explore. But it's very neat and all the same simplicity before.
That's like a skip list of sort, right? So let's go ahead and get started.
Cool. This is truly the foundational technology to analyze almost all of its own data.
Well, some of you know where it comes from. So we can split approximate nearest neighbor algorithms into three broad categories.
Okay. That is the first thing to tell you.
These are all part of a class of algorithms called approximate nearest neighbor as opposed to exact nearest neighbor.
The approximate is what makes it pretty radically powerful. They're all called a and an okay.
One more is called size by the way. So Facebook has an implementation.
It's called face for AI s face.
Face faces a kind of an okay trees hashes and graphs.
Now agents of Uber was agents of your graph.
So we can figure out who's probably belongs in the graph category.
And more specifically it's a type of proximity graph, which is simply means that, uh, start block number 11.
That is. So you can rapidly skip the link where you need to search and narrow down during such distance,
reduce the same skip list you saw before with these high degree vertices.
All right, you guys, we are less likely to get stuck in a local minimum.
And that means we're not going to stop early. So this is a full dense data meaning that is ultimately what you don't really want to search.

You know, using like full brute force search.
Instead, you initially search at a pretty sparse representation that quickly zooms you in the way you need to focus.
It's like a hierarchy, you know, like a quadtree. Okay. By the way, I keep talking about quad trees, right?
Look at this quad tree. I just say quad three.
Quad three. Data structure. Here it is. So all this notion there is a quad tree if that is your dot your index.
That is what the quad three looks like okay. It is always division by four.
And when you get to empty space you don't divide empty space over and over.
It is not a balanced quad three at all. It does not have balanced. It would be highly imbalanced.
But that is where the power comes in and only subdivide where you have to subdivide.
So it's pretty fast okay. Okay. So now we can go back and finish what you was trying to say a little bit okay.
Not too much. So that allows us to start the top layer and we move through.
I'll graph and we'll see. You want to get to the blue circle, right?
How quickly can you get to the blue circle?
We keep moving or traversing across different edges in that layer, in essentially the same way we did for an exact quote.
So from here, because that is over here, right. That's closer vertex.
So got our has and then jump and then distance from our query vector.
And we move to that. And then we we keep doing that. So we'll just quickly we'll get to the bottom.
But then once we we don't stop the whole thing.
It's very neat. So please take the time to actually watch it with, you know, sort of this valley of a classic data structure.
You know, you would know the distances we land around, but then they all become exactly code like M, if you remember, that's B, okay.
So all the vectors you load from this little vector file and the queries can come from that query file.
Now you do a similarity search okay. Number of neighbors that we and we see that one number is empty.
Right. Because we're building the graph we saw before the syntax reasonably intensive pretty fast.
And it takes a little bit of time I think maybe around 400.
I'm not sure of the the level so much of this code you've seen before from other machine learning,
um, you know, classes and so on, something a little bit new. Okay.
So these understanding articles are highly recommended because if one of them doesn't make sense to you, look at another one.
Same thing. Similarity vectors. Vector space cosine similarity.
Euclidean over and over. Same idea. Right? Underpinning small works in a way.
Okay. Skip lists how it works after a while.
Start to get it. Sometimes just animations in order is useful.
Okay. What else should I tell you? I should tell you about, uh. Uh, electrode, uh, in here.
Okay, so we talked about Mel was right. So now we should try, like, one more.
We should maybe try, uh, chroma. Chroma.
Chroma vector database. Chroma is also simply pip install chroma.
It's very cool. They even have Colab demo, right? So Jupyter notebook.
You don't have to install anything. You know water embeddings. It goes back to that same idea again and again and again and again and again.
That's all there is. Embeddings and then similarity property.
Your query is also embedded. And look around. So here they have this data set like your data set like this.
You should do this right. I mean if this is actually running for them.
Yeah. Oh you can actually run it okay. So like step by step you can run it.
Okay. Cool. Yeah. You know, obviously I'm not logged in, so you need to actually do it properly.
I would if I were, if I were at home, I might transfer this to my own drive and actually do it for real.
Okay. Anyway, so you see that gives you an example.
In the end this actually what happens, you know, what is stored food in a seed called it is called an endless stream.
That answer was not in the LLN or it might not be in there.
So you can make a biology specific but hey, this is funny.
Or this pastor. In fact, Sunday when was and the day before yesterday we had a Trojan hacks.
Okay, uh, a bunch of you got together and did, like, a little hacking.

So one other implemented an alarm rack application for CSE 585.
Okay, that took a lot of my lecture notes and made one of these.
And you could ask it explain MapReduce to me, and it uses the words that I use to actually explain MapReduce, which is so funny.
It didn't research, okay, you're going to replace me someday. So that is the whole idea.
I can make a little professor. Okay. I mean, seriously, this is a very cool like damn powerful.
Your homework. I'm going to give you three different homeworks A or B or C pick one.
And after the class you can do all three of them. One of them is something like this.
You would actually go in the jeopardy data set and actually, you know give it like a topic like physics.
You might say, um, tell me something, something about electricity.
It knows to go in the physics part of the, the questions and actually bring the answers back to you.
Okay. Whether I said this jeopardy search, but none of the words in the search, you explicitly have to train, you know, the nose with drag.
Okay. So that is what was I was hoping to show you a tiny video.
So, you know, give me a minute. 755 uh, chroma cloud.
Yes. Okay, so one more time, right. Look at this. You're the queries.
And the queries will then go to a bunch of vectors which you see, like right there.
And only then your queries answers that the vectors give you would become part of the query, which is called expanding the context.
Then now the query is now become augmented. You know, that's a little weird, right?
You ask a question which I don't know the answer, but the alarm goes to the external database,
brings the answer, but makes the answer be a part of your query.
But you don't know that though, because I'm just going to summarize it for you.
So very weird thing philosophically, right? Wow. Okay. So then expanding that okay.
It's neat. So again this filtering integration this is so cool.
This means you can use these programing languages to actually build chains of these.
And then you can do like more neat things okay. So in JavaScript it is so easy.
Node npm install with python import from adb instant.
When you say add that is really vectorized and you can print the vectors and you will see numbers.
And then when you query, that's also a number you can see and you can actually manually show you the similarities.
Character node okay look at this. Thank you. All this from example run okay.
I think we can move on. Uh. Of course. Similarity.
Yes. So let's see what this one is. Well, I already showed you this.
Okay, so pinecone has many. So this one is a little bit different though.
What is similarity search.
You know this is what the product you know people like Amazon would love because on Amazon right now you want to go search for a shoe.
You would type words like black tennis shoes right. But what if the person that uploaded the picture did not exactly use those words?
The product would never come up. But with this, I can take a picture of issues and centimeters just like that.
It is really powerful. Okay, so gradually people like Amazon and offer up, you know, order in Alibaba.
They all want to move away from text based search to some kind of image based search.
So very cool. So what is an embedding it also word of embedding is by the word vector is one of the earliest embeddings okay.
So in NLP you had this in a whole library called word to vec.
Select this vector embedding. So if you take NLP classes you will actually know about this okay is all.
But that is the key of all distance between vectors okay. You can actually do like Manhattan distance which is go along x and y.
In this case distance is two. In this case distance is square root of 21.44 column.
Or you can do cosine of normalizing it just cos theta.
You can use other distances also. Okay. There's so many different searches that's all.
And the nearest neighbors is your query. Your thing that you asked is here.
And those are the similarity. That is why the indexing helps to get to this really fast.

You are not going to do brute force search. You have billions of them. This is not today also.
So these things so beautifully explained from the people that actually do this, you know, for a living.
So you should read it. Okay. The built in and structures in, in all colors.
Yeah. I mean the basically give it away. So you your job is just to watch it, read it and understand it okay.
What else? What else think? So. Well, yeah.
So, you know, last year we used to call it infinite memory, but now we don't call it infinite memory.
We just call it rack. It's called infinite memory because the limit self has a finite context window.
Because attention can only be quadratic, it cannot be pretty big.
So we call anything that is not part of the quote realm external memory or so-called infinite memory.
But now that term is a little bit weird. We didn't call it infinite memory. We call it rack retrieval augmentation.
Then we use generative QA. So that means you can now answer QA by going to an external source.
And then your answers might be like highly accurate. Um yeah.
So let's try this one a little bit. I show you random things.
You can see new world. It's all cracked in all this, right?
Uh, moments of a surprising Q&A. So GK is what that is called, generative Q&A.
Say like this. So now the air has become like more cool.
Today we're going to take a look how to build simulated question answering app using OpenAI.
Look what happened, right? Look what I did. Check this out.
Today we're going to take a look how to build a generative question.
In that. So the NLM did not answer this.
If somebody has high quality answers about actual ML in our terminology ML lectures, then that can be used to answer this question.
So you're always going to get high quality answers, so you don't have to wonder if that's lying to you or not okay.
And in this case it is perfect. Wow. So maybe they could have said PyTorch is at a high level.
TensorFlow is lower level okay. But overall it is correct. So rest is totally right.
Oh okay. But how did I get that using OpenAI and pinecone.
So what I mean by generative question answering is imagine you go down to a grocery store and you go to one of the attendant and you ask them,
you know, where is this particular item?
Or maybe you're not entirely sure what the item is.
And you say, okay, where are those things that taste kind of like cherries but look like strawberries?
And hopefully the attendant will be upset. You mean cherry strawberries?
And they would say, okay, you just need to go down to aisle two and it'll be on the left or they will, you know, take you there.
That is basically what generative question answering is.
You're asking something, a question in natural language.
And that something is generating natural language back to you.
That answers your question. Now, I think one of the best ways to explain this idea of trying to check that out is to show you.
So we don't have that much data behind this is a six, a little bit over 6000 examples from the hugging face.
I thought you said hugging face streamlet forms, but it is enough for us to get some so much broader and heart.
So what I'm going to do here kind of play it twice to shorten, to sort of get a paragraph about a question.
And it sounds Irish, partly is hugging face difference or what are the differences between TensorFlow and PyTorch?
TensorFlow. I'm going to ask you this question. We can limit the amount of data that you can see.
That is where the top k k is literally how many nearest neighbors okay.
You can actually limit that by addressing tough case at the moment challenge written here.
On a very different note. Okay, for one more tangent okay.
Why the [INAUDIBLE] not? Today's a day of tangent. Okay. Um, if you want to make a UI like that.
So how the heck to say I'm a slider? I want a slider. You know, use type pi.
So type pi is probably the very best thing that you can do where you have a rag running on the back end.
But there are these things like temperature where how much can a randomized man set okay.
It's called temperature. Or in this case you know k right. So type pi kappa.
So incredible. Because it type II can make incredible UI like all of this, right?

And also actually productionize it.
There's something called Streamlit, and Streamlit is wonderful, but Streamlit cannot scale up type.
I can actually scale up. Okay, well posted video, put it on the stack and play this video and go back to that picture.
Okay, I never struggled to build production ready web applications using Python seriously, due to the complexity.
If you do any backend ML, which is all of the ML you do doing your homework course and everything,
please make a nice UI in type and you can deploy this on like all kinds of servers.
The world can use it okay. If the front end is back in development.
Type II is an open source python library for building your applications front end and back end.
You take the UI that you want and make a Python string out of a string with the word scroll bar field text,
field strength, and the Python string will get converted to real UI by type P like magic.
On one hand, it provides a simple and low code syntax that helps accelerate the process of
crafting interactive and customizable multi-page dashboards with augmented markdown.
Can you contribute something else to that? Formatting interfaces without requiring any knowledge of web development.
And at the same time, SciPy is designed to build powerful and customizable data driven backend applications.
It provides intuitive components to organize and arrange data through pipelines and data flow orchestration.
Type II provides a unique functionality scenario managed by TensorFlow, both data scientist and end users to perform what if analysis.
Configuring your pipelines and setup care for a studio.
The graphical editor. As a data scientist or developer type, I help you be successful with your Python development.
Whether you want to develop a simple pilot or a full scale application, either on Ides or on notebooks, it shows all the functionalities you need.
It has been designed. Okay, that's actually super cool series.
So they have made so many easy that you can try all of this facial recognition.
And I just try them all and then start modifying okay. Okay.
So when I apply just a little bit of this and come back information. But I think Streamlit the dimensions it's an older version so don't use it.
There's also something called bento box bento lunchbox bento and ML.
So bad times Streamlit cannot be replaced by type by. I should be enough for this.
So let's go. Because that's the neat part. It won't drag. Okay.
Uh, two the most. Two into some sort of car idea.
That is crazy cool again, right? So what is this object in the store that can be converted to text and then go and be able to answer you?
In fact, let me go back here. We can think of it as a model's long term memory.
Okay, so in the shop assistant example, the shop assistant has been working in nuts.
So while they have long term memory of roughly where different items are,
maybe they're actually very good and they know exactly where different items are.
And they just have this large knowledge base of everything in that store.
So once we get past that, a faster question. We are producing our language prompt or query to that person.
Something in there? Yeah. Something in that what we can think of as past experiences.
So then say again, same thing over and over. GPT is not answering your query.
It is going to an external rag that select a database just to get the one point in your head it'll be right next.
So to me, what is magical? Another. Okay. That's textual.
I'm sorry. That is your query. And then that is your vector database.
And your query is going to be innovation. What blows my mind is that this basic idea can be used for video search, image search and audio search.
And you can search. Right. The drum samples I showed you in the beginning part of the class could actually be like that.
Imagine a musician having all the drum samples vectorized.
They play the actual drum, produce me from the sample.
Something that sounds like this. That sound can then go and get something more higher quality.
You know, that's a loop, right? From what I just played. Like, what the [INAUDIBLE]?
Stuff like that does not exist. Okay. So that is what you should be excited by.
The dimensions, by it eventually. Now, one thing I'm going to keep going this way called like whiteboard painting.
Right? But then he's going to show you right there. Yeah.

So please walk through all of this. You can do it by one index.
So all about index. And the more pictures more and more.
And that is where this generation model is. Oh you saw the prompt go back to the prompt that you put in.
There's just one little part of the problem. The rest of the problem came from the tool augmentation.
The problem got bigger. And then your answer would then become amazing proportionately.
Okay, we can finish right on time. Okay. So only 2 or 3 more slides.
LDA stands for latent, which means hidden. Dirichlet Voronoi polygons are also called Dirichlet tessellations.
Yes. Right?
Yeah. You know, as far as I know, they never vectorized it. But something along those lines, they're trying to basically, you know,
pick out like whatever frequency are singing in whatever words, you know, but not explicitly vectorize it.
I don't know what they use, but not really. Vectorizing if you were to make a Shazam for two, you would 100% vectorized.
Yeah. Then you can go further. You can say, I'm humming a certain tune, so bring me other songs in a very similar tune.
You can know you can do Shazam Plus plus, right? Yeah. Which should do it.
Okay. So Dirichlet tessellation there is that I'm going to again amuse you.
Okay. Dirichlet. Tessellation.
Voronoi polygon. I put it on. Season.
Season? They all mean the same thing.
There's one more called brilliant, but there's a little bit more crystallography.
Okay. But why the [INAUDIBLE] not? Below in.
Low end zones. They all mean the same thing, believe it or not.
Okay, it's crazy researched and misspelled. Okay, I. E before I exoplanet competition.
Okay, but see, it is crazy that this data structure is so wildly amazing that so many people have called this the same in the same thing,
in so many different ways. Okay. Yeah. So this allocation is simply a topic modeling thing that I wanted to show you.
Otherwise it is all about classification.
So once you classify a whole bunch of documents and documents have this Voronoi boundary between them, then one new document comes in.
You will know exactly what class or even classes to assign. So one technique for doing that is called LDA.
So I'm going to show you LDA. How to generate.
An LDA topic model for text analysis. So in this case, what they're saying is that the new text is not coming yet.
But how do you even create one of these?
Like how do you train, you know, a bunch of documents to, you know, basically like no one categories of belonging.
So after this is done, you can feed it new text and get recommendations, classification, all the usual things.
Okay. So you can read this afterwards. But I wanted to show you uh, yeah.
So when you use Python in our read data again you can run all of this.
There's this cool library that some of you might have used. It's called Spacy or Spark C, as some people call it.
Spacy gensim. Yeah. So look here.
See ten best foods for you like delicious food sentiment.
This is sentiment analysis, right? Data cleaning.
Tokenize. This is all a classic NLP steps stemming.
You know, maybe even add lemmatization all of it. And then spacy.
Cool. So spaces what you can use for doing the stemming. Cut off the extra parts of the words.
Okay. Uh, create a document word matrix. So this is where it starts to look a lot like tfidf.
Okay. You have a bunch of documents.
You have a bunch of possible topics that the documents could be in and make a nice matrix, word document or topic.
I train a model to recognize that similarity. Okay. Cool.
So then just keep on doing all of this, right? So. Yeah.
So in this case, they're building the model by this library called sklearn scikit learn.
So scikit mix net. You know I think a lot of people use things like TensorFlow and then Keras right.
And then maybe even things like PyTorch is a cool.
They're all used in production. They're used in the industry. But there are two more.

One is called MXNet and one is called scikit learn.
Scikit learn, sometimes called sklearn. These are also pretty amazing.
It's a little less sexy, but they're also so robust right by so many people and over such a long time.
So sklearn the scikit learn has a LDA, you know, a function.
We can use a character like duration allocation. It is that simple.
In machine learning courses. Data mining courses. We will teach you the math.
Go and do linear algebra on a board and drive all of that. Okay. But in no way I'm gonna tell you it's a waste.
Okay. It's nice. Why? You would rather know in one line.
How can I call it? The world will pay you for this. The world will not pay for your dot product knowledge.
Okay? So always good to have, don't get me wrong, but to use it.
It is that simple. Select that model free transform.
Whoa. So cool. Okay so I'm going to keep going. And then now we basically evaluate a model right.
But then once the model is done. So now new documents can come in and properly get classified okay.
You can go through this afterwards. But until you see like you document belonging topic you know highest contribution.
So when a new document comes in the document can potentially have all the words meaning all the terms that you chose in the document.
But sometime has to be the most frequent term. Say the document is about LA.
Yeah, they'll talk about cars. Also they may talk about stars or something, but mostly it's about LA.
So find that one dominant terminal case with a single hot.
Okay, so like here. Documents go this way. Copics go this way.
Not all documents are equally strong in all topics. For example, this first document seems to be about topic number nine.
Uh, this document over here seems to be about. What the heck?
What is this? Okay, so this particular document doesn't have any other topics at all.
And that's totally okay. And you got to randomly pick one, right? Yeah.
Okay. So it seems to vary obviously for document number ten.
Again that last topic we get the idea okay. But there's still one dominant topic.
And then the idea is you know when a new or new document comes in you should be able to pick the dominant type that isn't that is all you guys.
So that predicate topics. So once you built the model, once you have the training done then the new text would come in and then do this,
do all of the things to a new text like a new query vector, and then also do the transform and then see where it lands.
So a lot like all these things that only occur. Great.
Okay. So then. Yes. Um.
Oh, this is actually very cool, right? Okay, so if you have a, um, like a customer interface, people go and complain about your product.
One of the first things a company wants to know is automatically look at what user type and ask, what are they complaining about?
Are they complaining about a product or the delivery of it, or some insurance in order to pay for it?
Who knows, right? So knowing that alone is enough, because then you can route the complaint to the appropriate person that's going to handle it.
So look how useful that is right? So you need to basically classify customer complaints as to come in.
And pretty soon in real time while they're talking, to get the right person to help you.
All right. So then that is all. This is actually very cool, by the way. So I draw lots and lots of clustering diagrams.
Right. That is for real. That came from the actual data set we used.
Clusters are real meaning this is like one set of topics, second set of topics.
And obviously there's some overlap merging going on but are still pretty clear boundaries right.
So that is ideally like what you want. Okay. And then this is the recommendation engine.
So I'll give you a piece of text. And you tell me things that you're having a database that are pretty similar.
Turn it around okay. Need and see what they do, like Euclidean distances over and over.
Same thing, except you don't have to calculate it. You can just simply call some kind of distance formula say that the greater.
So you don't have to do the square root of x squared minus y squared. That function can do it for you.
It's all done for you already. This is like so cool right?

I think these are really helpful. And what other documents can you find really helpful.
Well to similar documents. Well. In other words, find other documents that have all these words in it.
It clearly works. Okay. Neat. Huh?
This one. I can go a little faster. I already told you. Approximate nearest neighbors how to classify all these vectors.
Okay. Suffices one of them and their face. And F stands for Facebook.
Why approximate? Because I'm much faster than the real one.
Yes or no? Oh, okay.
Well never mind. Can find other articles. Okay.
But what about this one? Yeah. Comprehensive guide.
Oh, so this one is a guide about all kinds of algorithms.
Right. And one second very similar to what I have. So all about the same idea okay.
Um so nearest neighbors motivation and let's use case Netflix recommendation TikTok.
At some level that's what all engines do. They know what you watch.
And then that is all embedded in like a query against all the billions of videos that they already have, and find out what kind of videos you like.
Cat videos, okay, baseball videos. And then start lining it up and they're going to like it even more.
So Netflix, Spotify, Pinterest. Cool.
So many of them do the exact same thing. And then exhaustive search.
You can also do exhaustive search. By the way, that is what the actual vectors look like.
The pure floating point numbers. You can see they actually points in multidimensional space right.
So you can go this are quite long article. I won't bore you with all of them and walks you through okay.
This one is brute force search okay. It cannot scale it all. Then they tell you what can the N and algorithms can you start to use.
That is one thing like size and all that coming okay.
Others vectors. Now this is all the vectors and that is all that in the data that I keep doing all over the place.
Okay. Oh, annoying is also a library. It's called annoying, but it's actually ten and stands for Nearest neighbors.
Okay, so you can use the annoy library if you want. There's so many libraries, you guys.
All right. Again, the same movie names in there.
Vector encoding. Locality sensitive hashing, which is another name for this kind of spatial index hashing okay.
So you can do spatial indexing. So right there and then LSH uh quantization.
So many different ways of doing the same you know search of clustering.
So look at this. Uh, this one is actually, uh, K-nearest neighbors.
Okay. Okay. And k nearest neighbors at all. You say you want three nearest neighbors.
You say, give me three clusters. Right. It doesn't know where the cluster centroids are.
So you randomly start somewhere at the bottom left actually. And they all migrate to where they need to be.
So I showed you the JavaScript enumeration of that right. It's going to kind of.
All right. So then this saw, like, vectorization stretching you.
Okay. Then they all become quantized vectors. All right.
What else? Just to finish it on time. Oh, okay.
This one is actually very easy. Task specific fine tuning is just simply you can take things like GPT two even, right?
It comes already with the large measure text corpus you train it on. Yeah.
I was very sorry about. Okay, I'm going to stop you, though.
If you just hold on. Because then I can finish and others can go. We can definitely talk.
Okay? Uh, looking at him and saying, let him finish. So I'll definitely finish.
And on top ten. Alright you guys, so you can take Gpt2, right?
But also take 100,000 scientific documents and do the fine tuning.
It's all there's no rag anymore. Now that I am becomes capable of answering send a request, then we can call it task specific training, right?
That is basically what this is. So fine tuning is one alternative to rag.
In the real world. You can do both. By the way. You can do fine tuning and rag, but this one is about fine tuning.
Okay, go read it. Just simply put up like the whole article.
That's really all you can read. And then you can run them actually like Python code.
All of what I tell you please, please, please run. And what is crazy is you can generate abstracts.

So you want to write a new paper. The whole idea here is make me an abstract and I'll make a paper out of it.
It's kind of weird actually. Okay. But the abstract is generating is not bad.
Okay. It's giving you some ideas for some papers you can write. Okay, so last but not least, as they say.
Oh, so one more task specific embedding.
You know, you can go through this yourself.
But this again goes towards agent based programing where you don't make one giant and AI embedding that does it all, but do it for like one task at a time. So I throw like a lot at you.
Right? But which is okay, but hopefully within the next couple of days or something,
you should have some homework, three different versions of them that will cover like a bunch of these.
Pick one new, about two weeks to do it and will be all done. Okay, see you next week.

Lecture - 5
And I think it's gonna be a long, long time coming.
You gonna get to find my mom?
I'm gonna call my mom. Mom? No, no, no, I'm.
I. Love it.
Love you. Love him all.
And I think it's gonna be a long, long time.
And I think it's gonna be a long, long time.
And I think it's gonna be a long, long time.
And I think it's gonna be on. No, I.
It's all you were rolling. And I think it's gonna be all.
And I think it's gonna be a lot of long time Elton John.
Sorry, John. Thanks for the.
Oh. Hey, we should just sing songs for three hours.
When the Beatles don't come. I know that you're.
You know anybody that wants to sing during the break? We still have this week.
And we still have next week. Please do it. Okay. You know, I have way too many favorites.
I mean, this is hard to pick on. No. Every guy trying to win.
Are you going to come and perform? Oh.
Then stop. The music stopped. Oh, it's like musical chairs.
Oh, non-data. Oh, no. It's 24.
Such. Wow. Look, a clean board.
Two weeks, we wind it all down.
So, uh, next class, three hours, and it'll be, uh, all wrapped up, I hope.
But I have a homework for announcement to make.
As I told you on Piazza after this class tomorrow.
Really? I will definitely put it out. Three out of four I already have done.
I prepared them for you guys. Some, you know, three already. The fourth one is almost ready.
But I'll go home tonight. Finish it. The fourth one is the one with the lightning I.
So all of this or one way or the other. But let's.
Search is absolutely headed that way. Wait till I show you.
Oops. Can we switch to our screen, please? Let's talk on the WebEx screen there.
Interesting. Laptop switch, please.
Okay, so all that happens I was talking okay so I'm really are what are going to power search and so much more really at some level it's all data.
At some other level it is all search I what's happening what's happening when.
No it was stop. So I just have to reshare okay.
I'm not sure if I did it on purpose. Okay. I don't think I click on anything or maybe accidentally click stop, I don't know.
Oops. Sorry. Okay. Now I know what to do.
I will not click on stop, I promise. Okay, I will stop, but I.
All right, so, you know, I love search data. It's all, like, wrapped together.
Wrapped up together. So after four homeworks, three the four little questions, three of them are actually about lamps, meaning they'll use an alarm for this one and an alarm for this one and this one as well.
So that's already pretty cool. Different ones. This one actually.
It'll create an input meaning what you would do is create like an input for an alarm.
So that is not directly an alarm question. But if you want you could take that input and even for future for fun,
try it out here and here and here if you want, because it is nice to even practice.
You know how to do that. Okay.
How are you going to generate, say you work for a company and the company has all kinds of, you know, documents that live on a server somewhere.
You have to collect them all and make them be the input for a future.
Lambda is not being written. So it is good to practice that. So all the forum, right are all related in that sense.
Hey you guys, please, please don't talk. There's only two more weeks, so let's keep it quiet please.

When you talk, by the way, it actually bugs people next to you. You know, if you don't realize that.
Um, yeah. So you have 12 days, which is actually great.
A long, long time to finish them. Each one will not take it three days.
A promise probably won't even take you three hours. But don't put it off, okay?
Just because I told you that, please don't make it a low priority. Uh, you shouldn't ask for extensions.
12 days is a long, long time. All right, well, basically, make it be do at the end of the terms.
You should be. Okay. And then there will not be a first homework, in case you're wondering.
You know, there's no rule that says any course should be exactly 100 points.
You know, just simply a nice number that we like, you know, many undergrad courses or for things like 1160 points.
So philosophy course might be 1184 points. Okay. Why.
No good reason can be for anything. So then pretend that this course is not for 100.
For this 100 minus, you know, whatever it took it. Okay? I'm not going to change the percentage of anything, so don't worry.
The homework is not suddenly more important or less important. Everything is the way it is and it's all relative anyway, so nothing should matter.
In other words, it's almost like a give all of your full points.
For the fifth homework, go and fill in the column with exactly the 15 points or whatever.
But that's pointless, you know? No need to do that. It's the same effect.
It's all 100% relative. So? So no one should worry. Don't be concerned.
And then for your final exam, it'll be a lot like the midterm.
It'll be in a classroom. But this time we have 950 spaces.
There's only about 450 students. So what happens for midterm won't happen again.
The midterm, by the way, was not really our fault. Just say no. So we didn't ask for this class.
Actually classroom scheduling people give it to us. But this time we're very proactive.
We already got the rooms and they're like spread out in this whole area.
So everybody would have nice adequate seating, you know, so should be okay. Everything should be okay.
And you can use a cheat sheet as usual. Like for the midterm.
Except for finally. Hardly need anything. Is everything that I tell you in all these lectures.
You know, I got, like, a bunch of things.
On that note, I did put up the slides for the things that I forgot last time, you know, and I canceled the class two weeks ago.
I forgot to put up the slides for things like question answering. So one of the students asked Ashton Piazza to put all that up.
So here it is. In other words, these slides are exactly what are covered in those lectures.
Okay, so between the lectures and the slides you should have everything in there.
And the exam is going to cover mostly the second half.
Although obviously you cannot forget the basics like where does ranking, you know, bunch of documents mean, you know,
just not as clean break in your in your mind, you know, so go back and look at maybe a little bit of the first half, but it's not evenly distributed.
The exam points will not be 50% for the first half. 50% of the second half will be mostly second half because I have enough to ask you.
Cool. And then one final change I want to announce is here next week.
I'm going to call it Assorted Topics part three. You know there's so much change.
It is bleeding edge. You know what is happening. Uh, okay.
This rag. Correct. By the way, some of these rag questions like this one might be actually right.
Question. So you're going to practice rag already.
This rag two point hope you know as if all the previous rag was called a rag 1.0,
Microsoft basically came out of nowhere and said, no, we'll fix rags problems.
Rag also has problems. So then that's called Rag 2.0.
I have like a nice laundry list of topics that I want to talk about.
Those are not covered here. And you know, in part, part B of these assorted topics.
So I'm going to make a part C and I'll wrap the whole legal aspects.

So the legal aspects of all of these are highly out there.
In other words, it is all not just only entirely good news for everybody because all kinds of very questionable things, you know, CEOs come and go, you know, lawsuits, okay. There's a lot you have to know all of that also.
Ah, the ratio would be blinded into thinking this is magic technology that solves everything.
It's not true at all. So I want to cover, you know, what the problems with all of this might be.
And there are lots of problems. And maybe some of you can try and work on solution zero.
And so, you know, talking about problems is valuable. You don't ignore the problems.
But I would definitely add that to a whole bunch of other things I want to tell you about.
Okay. So I'll call them all part C. So between part I, part B and part C is really basically everything that's going on in such clear.
And there's obviously no textbook, there's no blog post, there's no LinkedIn group people.
It is so new. Um, also, there's no sequence of topics to cover, like at least here with information retrieval.
The first has the older half, you know, you could start somewhere and gradually work your way up to things like inverted indexes.
With this, there's really no fixed set of topics really. You can jump in almost anywhere and work your direction through.
And that is what Bleeding Edge actually means. Sam dumping you into the bleeding edge.
Okay, so I hope you think it's okay if someone taught me a course, that's what I would want them to do.
I don't want them to just stop with anything before all this and say, of course is done.
Then you have this partial view of the class, except you don't know that.
So when you go in the industry, people are using all of this. In the last year, no one taught you that, you know.
So I want to be sure that I just want to, you know, talk a little bit here, you know, so this is basically education, right?
Leading up to like when you graduate with the masters or bachelors, there's definitely a sequence, you know, starting from young preschool.
But at some point you graduate to go in the real world. Okay. But for a lot of people, the real world, right.
Real life, real world has a pretty steep step after you graduate, which means you join like Amazon and they give you, you know, a good project and say, here, just pull the repo and make some changes and send like a pull request to the owner.
Like, what the [INAUDIBLE]? You just say, if we never talk to your git, right, you have no idea what the [INAUDIBLE] that actually means, okay?
Or we'll teach you databases. It's all relational algebra on normalization.
And then somebody has an actual database here. Please add these rows of data to a SQL query.
I don't know how that's a pretty bad thing okay. In other words, industry uses nothing but tools which are simply implemented versions of all of this.
So you take all this theory, for example, inverted index, and then you turn that into an elastic search, you know, interface Java API.
And that is what a company would use to build a search engine.
So unless you know to use this all of this knowledge literally just a pure theoretical knowledge.
Right? I'm going to call it a pure waste okay. Nobody pays you in the world just because you know a bunch of things.
I mean, nobody cares.
Okay, you said the reason why they would care and pay all that money is because with all that knowledge, you can go and use the tools that they have.
So my homeworks hopefully try and provide like cream for you so that it's not as steep.
You know, it's more of a gradual change. That's what I'm trying to do.
That's why all the homework show a practical, you know, that might not be very complex.
There are not very challenging research questions or something, but just by you doing them, you will know the steps.
And by doing it three different ways them you will fully understand, okay.
And that should be enough to for you to go in so many purposes. So that's really the point of me, I'm doing all this for you.
So therefore part C will also be a ton of topics.
That's all. Okay, so I'm going to go in, um, dexterous for a minute and then show you something very cool.
And then we can get to our topic today.
This one chat do while it's actually very cold, is sent by one of our students, Karthik, who was also MIT for a graphics class.

Very cool. So what this does is you can actually this by the way retrieval, augmentation.
In other words, say you want to search through repository documents.
You want to ask things like what are the top ten react components.
Do that right now. It will actually then go in the documentation for those repositories.
There's so many different repositories that many, many, many of them.
Okay. Search for repo. So you can even search for anything that is not here.
For example you can search for like, you know, view or something, right? Vue.js.
This is by the way I tried about an hour ago mean I've been using this for a while, but then about an hour ago it quite didn't work.
I think there's some something. Something is down, but it's not bad, right?
It's back. But say you picked that. Now suddenly you can actually ask some questions, you know, based on the repo.
Uh, okay. I'll just pick something here. Let's pick svelte.
The idea is you can ask select here. Right. For example, how do I create a component?
See that there's something wrong with the server. But maybe try it tonight or something.
It'll work okay. It's super cool. It'll answer your questions.
Now simple questions. Very detailed questions. By going through the actual documents for the repository.
It means you don't have to read the developer documentation. You can ask plaintext questions when you get stuck.
It's a little StackOverflow. So cool right? That is an example of what is called retrieval.
Augmentation. It is really cool. I mean, how practical this whole drag idea is.
So I wanted to show you that, but sadly it's not working. But it's great though.
You can even add your own repository and then your own documentation,
your colleague wrote, and then browse the documentation using natural language questions.
It really works, by the way. It's just cool. It will never hallucinate.
Hallucinate means the core team does not know anything in detail about Reactjs.
So if you ask it, you know what are the top five components? It will come up with random words.
Okay, from some react documentation, it's a complete lie and you cannot trust that and say, wow, I think I'll go learn the top fragment.
It's a complete lie. We call it a hallucination. So this way it avoids the hallucination.
That is basically what this whole chat do is. It's so cool.
All right. Uh, so that's like the non hallucinating highly contextual up to date because it is not storing anything.
It's actually going in the repository in real time and then querying all the documents and getting answers for you.
It's neat. Like that's exactly what you would do. That's a chain programing language and chain programing language.
But see, it's not working. Okay. Next one is so cool.
This one is you know what can happen. The search engines including Google okay.
So search engines initially when they are released they have a noble aim.
They want to serve knowledge to the world and organize all the world's knowledge. But gradually, over time, they become corrupt.
Like people can become corrupt, they lose their mission. Okay, so to speak.
They lose their way. And then people then in turn says, screw that, I'm not going to use it anymore.
And so, you know, that actually can happen to Google, believe it or not. Okay.
In fact, I can actually, sadly, lead to that. In other words, Google was the undisputed king of the hill all these years.
But now suddenly the world one school to summarize the search. Slowly but surely.
And if Bing does it right, or some third party comes and that's it, even better than Google or Bing, they'll lose.
So it's entirely up for grabs. You know, the whole search world, which is worth hundreds of billions of dollars.
Think about it. So this one is so interesting, this notion of Shaggy.
So Tagi was a cool search engine that again, like, went by the wayside.
And so this article is just simply telling you, you know what the problems are.
I'm not gonna read all of them, but. Okay. Yeah. So the idea is that this can actually happen to anybody possibly.
Look at this. So you can try Kagwe if you like. All right.
That is why these days, you know, when you want to look for something and not be bombarded with ads, you're looking for something real.
Like, you know, what are some travel, you know, don't do is, you know, go to Saudi Arabia, something like that.
You need to then type whatever your query is and add the word Quora or add the word Reddit.

So that way the Google search will then helpfully give you a Coracao article from some real human being,
the raw textual thing, and you have the answer. Whereas if you leave it to Google search,
they'll start advertising random hotels in Saudi Arabia because you type Saudi Arabia the words the search is polluted because of money.
They want money. So people are slowly getting tired of that. Right? So it's pretty interesting this.
Yeah. How you got this? All right.
Um, so that's about Shaggy. What else should I tell you?
Not much. Really cool. So, unlike the other day yesterday, almost instantly we can get to it.
So I told you part a last time, and I'll finish with fine tuning. You know, I went a little fast, so we'll get to the last slide.
These topics again are sorted because like I said, okay, think of this core knowledge of anything in the world physics, astronomy, biology.
You know, we know so many things pretty well that are lost in our time tested theories about how people behave, you know?
But the bleeding edge is like a fractal, you know, it's evolving in all kinds of strange ways.
So nobody knows exactly how this connected to that. No, there's no theory.
You want to connect them. Like we don't even know if it's the right answer. In fact, let me tell you.
Right. It's cool and everything. Right.
Like 2.0 is cool, but a much longer view that ten years from now is a very long view is that Rag will become 100% obsolete.
What the [INAUDIBLE]? You've been playing of rag all this time, and you're telling me that it is true?
We don't need this thing called rag doll. Okay, so rag is there to solve a different problem.
It's about context window length. But when the problem is solved, actually Google solve the problem last week.
Yeah. Again you have to wait for next week. Let me tell you all that. But meanwhile rag is not really here to stay forever, so it's crazy.
You know how quickly things can change, right? So that's what I call a bleeding edge.
So bleeding edge of knowledge looks like this. There's no good entry point.
No no no no I'm not qualified. Nobody is qualified at all. You first do this and then afterwards do this.
And after this there's no laundry list. Okay? It doesn't almost matter where you start.
You'll be initially stuck anyway, but you will gradually figure it out. So that's how I approach learning anyway.
I don't look for formal entry points, I just jump in. Sometimes it's too much and I back out and you know, you give up.
But it's still worth taking this approach. So yes, the vector db is all that vector db, but you're not going away okay.
They're here to stay. But even their Euclidean standard XYZ, the square root distance is not the only measure for similarity.
I told you this manifold measure. So you can generalize it, but it doesn't go away.
It makes it more interesting, more complicated. So I talked about image search.
And image search these days is pretty much done by similarity, meaning all the images are embedded in some kind of an image space.
You know, now that I have a little time today, right?
Because the second slide is only like a small set, I want to explain some of like what I mean, I'm just going to take one image and explain it to you.
What do you mean by embedding images. How do you do image similarity?
What does all that mean anyway? I'm going to type the word sunset okay.
Just simple tabular sunset okay. So then I just want some image talk about you see that that one image right there okay.
That one pixel the very top pixel right there. It has one RGB value a singular value.
You can say. Likewise, if I take a different sunset image, that RGB value for the first pixel is not the exact same RGB as this.
You can see the color is slightly different. This one is more reddish and purple.
This one is more blue. Right here the same RGB is more orange, which means that it's more red.
More green. You get the idea. Okay, so just one pixel, okay.
Just one pixel from many million images. What can we do with it?
Here's what we can do with it. If I take if I make an RGB axis like this, I have an axis right here and I have a g axis and I will be axis.
Each axis can have values between 0 and 255.

Because there is so many values, a single color channel can have this also 0 to 255.
So that is 256 values. Here is also 256 values.
So total number of combinations possible are just simply 256 times 256 times two six.
Which is really to raise to eight times, to raise eight times to eight.
So to raise to 24, we call it 24 bit color is approximately, you know, 16.7 million colors.
Okay. That's all that is. How many combinations are possible for every possible value of RG and B.
Okay, so now go back to the photograph. This first photograph has some value for some value for g.
Some value for b. So I go in this axis and I plot that looks more blue by the way.
So maybe it's high in blue. It's low in low and red. So maybe the point is here in 3D space with a little vector like that.
Whereas the third censored picture is more red and green is more yellow.
So that's more and more green, but not too much blue. So it's floating in space, but then it's more close to the plane.
And maybe the point is okay.
So I can take 1 million images and take 1 million pixels, the top left pixels, RGB values, and plot them all on the same coordinate system.
I'll have 1 million points, okay.
1 million points that pretty much cover lots of the blue because lots of sky is blue and lots of red green, but maybe not too much green.
You will hardly find any sunset pictures with way too much green and nothing else.
Sunsets hardly look green. Okay, maybe Aurora borealis.
If you go to Alaska or something, you might see a little bit.
So there might be some images right here where the first pixel was green after all, but usually not.
But that is only for one pixel. Imagine then I do this for two pixels.
I take the first pixels RGB that is three dimensions, right? And the second pixels RGB three more dimensions.
Likewise first pixel. Second pixel six dimensions.
First pixel second pixel six dimensions six dimensions.
So now I add three more dimensions. I would add an r prime, a g primary prime, but they're all mathematically 90 degrees to each other.
We cannot anyway and imagine that it's like a six dimensional cube, but it's simply just mathematical there.
Axes, you know, like a vector, a six dimensional vector. So therefore between those two pixels RGB, RGB, it will become a value.
It will become a uh, an array with six values. For example, maybe the first color was almost pure yellow sunset and maybe a little bit of blue,
and the second pixel was almost all blue, you know, transition.
So maybe it went from 80 for red with 90 for green and 200 for blue.
That's two pixels. Okay. So that becomes a point in six dimensional space.
Keep going. Suppose you have 1 million pixels 1000 pixels by 1000 pixels like one megapixel in an image.
Okay. And that's 1 million individual pixels. And each pixel has three dimensions.
So 1 million times three is 3 million. So imagine this to be a 3 million dimensional axis, a coordinate system.
All the 3 million dimensions are 90 degrees to each other that are orthogonal.
So every entire sunset picture, the whole picture will become one dot.
This will now have 6 million values. Matter a but it's a 6 million dimensional vector, exactly like all the indexing query that we've done so far.
Okay. Same thing Tf-Idf model. So in this embedding, which is no longer three dimensional, every sunset picture in the world,
because they all look somewhat similar, would all become clustered in some part of this 3 million dimensional space.
All the cats, cats with little cat ears or all become clustered here.
All the dogs get clustered here, all the shows get clustered here.
All the blue jeans get clustered here. So the clustering is natural because almost all blue jeans look like each other.
Almost all shoes look like each other. So spherical, almost all faces look like each other.
So they can all be clustered in very specific locations where you can.
No way in [INAUDIBLE] imagine. Even God cannot imagine 3 million dimensions all at the same time, right?
But mathematically it's possible. So that is why similarity comes in.
If I then like, I should show that somebody is wearing one normal shoes, that I take a picture of the shoe and I search,

then what happens is that picture of the shoe, the 1 million pixels there will become a 1 million pixel value query vector.
And that that might end up maybe here, the show that I'm trying to search for ends up here one point, and there are so many other shoes nearby.
And guess what? They're all the same shoe that the person is wearing, which is all the way down here by each other.
So if I do a little similarity search using like a little neighborhood, I will faithfully retrieve exactly what that person is wearing.
It's amazing. There's no SQL, there's no code, right?
There's no text based brown shoe. It rings in the back.
Nothing. Just image search. So that's really, really powerful because anybody in the world can do that.
They can take a little picture. Right. And search. Great.
I can take a picture of my stuff in the fridge and ask the system what what tasty recipe can I make with this less than 1000 calories?
I can even put that in so it'll faithfully go get exactly what you want.
Okay? Because it is doing similarity search. So please don't forget the whole embedding okay.
Okay. So I simplified things a little bit, but it's it's okay.
You don't take the raw pixel values and actually stick them in here.
You can. That's one way to do it. But maybe a different way to do it is to train a neural network okay.
It's called an embedding neural network. And then make the neural network produce these vectors from an input image.
So this neural network that is all trained on all kinds of images, you punish it.
You rewarded punish it by backpropagation to recognize certain things like cats, dogs.
Then you want to, you know, embed like a new thing in your query. Then you can then take that new query and give it a new.
Network, and that will then produce the the vector the query vector.
And that query vector can then be searched against all the other vectors,
because all the other vectors were all embedded by the exact same neural network.
So can have this embedding neural networks okay. But in the worst case you can use just raw data raw audio values for all frequencies,
raw pixel, raw words you know, so you can embed sentences, you can embed parts of words.
You can embed a whole paragraph. You can embed like a whole book. It's all about embedding okay.
So the call embedding layers you can look for them. But the reason why I went there is to tell you about how, you know, this notion of.
Similarity all that we're talking about. So where where. But yeah image search.
So when you talk about image search that is actually how lots of image searching was done.
That's what Google does. Soon you upload an image and search with Google Lens.
They take your image and then they, you know, vectorize it, meaning they,
you know, turns it into one of these queries and then do a similarity search.
It works sometimes pretty well. Sometimes it doesn't work. Well, suppose you're walking by and there's a flower.
It's surprisingly accurate. You have no idea what the plant is called.
You can take a picture and instantly 100 different images come up from different
nurseries and botany gardens all over the world with the botanical name,
the Latin name, you know, how do you grow that plant? It's great. You don't have to ask a human being.
They might tell you something incorrect. Yeah, you can use it for searching mushrooms.
You can use it for searching and learning anything you want. Cloud shapes. Right. It's great.
So I talked about code search and I say code search is not anything different, but it's more optimized more tuned so to speak.
Just to do course you. As a very smart question.
I mean, yeah, the short answer is yes. So now you talk about what's called combined loss.
You know, I talked about loss last. Meaning what if it's not able to predict properly.
So then you can take an image like a sunset image and then now provide some text for it.
You can say, um, deep blue, purple sunset in a glorious sunset, or amazing blazing orange in a yellow sunset for yellow sunset image.
Then the words can also be used as training data along with the pictures.
Then that's called combined loss meaning then you give it an image and say label it for me.

Like tell me what words it is. If it produces wrong words, you punish it to go on training till it produces the right words.
Okay. So then it's a combination combined embedding combined loss okay.
That minimizes the combined loss. That's pretty cool. Then you can use it for all kinds of things.
Then if you give it an image it'll caption it for you as wonderfully as you would caption it.
Or conversely, you could type a caption. You could say, paint me the most glorious sunset,
and then it will become generative AI and then generate like a new image by combining like so many other images, but properly so.
It'll make your new images so, you know, the image generators work. Yeah. So that is definitely combined loss.
So you can combine words with pictures words with radio.
That is all very cool by the way, because then um, say Ikea product manuals that nobody reads, you know.
So then if somebody goes and annotates what the product manual is saying,
like lean the beam vertically and put this post in this little potential font,
then you can start talking to the AI and say, I have this Ikea bookshelf and I have no idea how to fix it.
What should I do? It will tell you, you know, pop up the tallest beam against the wall and lay the other one five feet from it is talking to you okay,
that is the whole bicycle chain demo that I made. And my bicycle chain is broken.
How do you fix it? Then you show a picture of the broken bicycle chain. It will tell you exactly what to do.
So those things are, like, truly magical. It's almost like having this expert, you know, that knows everything.
And then you have to ask it and it'll tell you. So it's actually useful. All right.
So chord search you know it's like highly structured because code is not all words in the English language right.
And so it's just a few keywords but code convert comments talk about combine loss okay.
So you can have humans annotate code. You can be paid to annotate code.
And then the annotation which is the text comments.
So what the code does the following code generates a uniform random number distribution between -10 and 10.
Suppose you write that. Then that.
Those words can then be trained along with actual Java code that produces the normal distribution between -10 and 10.
That becomes your massive training data afterwards. That's when generator AI kicks in.
GitHub copilot. You can go and type random numbers between -1 and 1.
And suddenly before you hit enter, it gives you the code in like 100 different programing languages to look at while
and in VS code you can say adopt and it will then become part of your code base, you know.
So that is how that works. You can do training based on text description and code, but you can do search as well.
You know, you can obviously do search based on limited keywords limited because again,
code is more limited, more structured than uh, just general text.
All right. So then again we've talked about this is very cool. You know here the pretty much built an inverted index.
That is all. But in the inverted index that the words that are inverted are simply keywords like for if function return in a while case.
And then uh, what is pointing to is actual GitHub GitHub repositories URL.
So it's exactly like Google search, but not any keyword, any URL, but more like programing language, keyword search and GitHub URLs.
So that's highly useful. It means you can go and search anything you want. I think I did the last time right means train.
Just go to github.com and you can type words like rust, memory safety or something.
And all kinds of crazy cool things will actually come up. Remember Karpathy slalom.
Next time I want to tell you about Amazon. I've been saving the best for last in a way.
Where do you search? There. Here you go. Okay, suppose I type rust memory.
Okay. Rust memory. Then those become the keywords in the inverted index.
And then they already had cache like 910 results. And here it is.
So something about that rusty memory load library.
On and on and on. Cool. Look at that.
So it found all kinds of useful repositories for me without this.
You know, imagine GitHub has many tens of millions of repositories.
You would have to manually look through them, you know, or imagine is doing some bad search not based on actual structured keywords.
[INAUDIBLE] give you relevant crap, you know, that is basically not even about rust, you know, or memory.

So this is perfect though every single repository, it finds a link to every single one of them because look, it's looking for the keywords, right?
Exactly. You know, like or inverted index. It's all I'll call.
Searches for location based search. I have more to say about this.
So location based search is where you type, like, you know, pizza places nearest to me or something, right?
So let's try let's say a pizza near. Then ask, you know what is going to happen.
You should find the approximate location through Wi-Fi three or phone line and start showing you Jefferson Boulevard.
You know take stick to it knows if I go in the Marina or I go back home, the location and type the exact same query, it'll give me places near me.
Okay. But right now, this is what location based search looks like. Or they might give you a map.
See here. Um, they tell you exactly in relation to where you are.
Maybe, like, you know, you are, I don't know, here somewhere. Then you can see in the area, you can zoom in on the pizza places.
It's quite useful. But where can we go with this?
Already Google has come like much, much, much more further than this.
So I'll tell you what I have in the slides and I'll tell you more. Yes.
Uh, so using the query, using the location of the query, it actually answer the response is extremely useful.
And this article some business in your daily just gives you like an introduction.
I won't read all of them for you. We did this last time okay. But I tell you, the more fun, the more the recent things that Google does.
Yeah. So what is it? Why is it useful to solve for a business point of view?
Right. So if you are a business owner, you should actually care about all this.
You know, these are all ways to market to your customers.
Um, if nothing else, like I said, your phone can even suddenly flash a coupon if you walk nearby a sales rack with clothes in it.
You know, that's how much they can know you. That is micro targeted advertising.
Your friend might not get the same coupon, but you would get it. So like, yeah, so many things you can do with just location.
This is extremely useful. So you are in an accident, okay. Your car suddenly died on the road or something.
You're mentally freaked out and in no position even to describe where you are.
You, like, panicking. Like somebody is going to come and kill me. Please help me. So then you are the last person to say what is nearest cross street?
Okay. The person can answer, kind of like panicked.
So if the car knows, you know, the high end car scenario, they can send assistance to you immediately, even without you asking them.
So that is exactly one of those uses. Likewise Uber, Lyft, all of those are nothing but location based services there.
They use stage three hexagon indexing for their passenger who request a ride,
and all the drivers nearby who are in that are in the adjacent hexagon and alerts all of them.
One of them picks up the right. So why should you know if you pick up a ride from here, alerts somebody all the way in like Long Beach, okay?
They are not going to drive all the way to pick you up rewards.
It has to properly match the drivers in the passengers based on proximity to each other.
That's exactly what location based service can do. So there are so many neat things in here, right?
I mean, even this one, I mean, that's one of the most obvious things. Okay.
If I have never used my credit card in the past five years outside California,
then suddenly there's a credit card, you know, notification that goes to the bank from England.
You know, that purchase should be pretty much 100% decline.
Or obviously, I could travel to England, but in the past five years or something, I didn't do it.
So most likely somebody stole my card and tried to use it. Ringland that's the easy location based thing that they can do, right?
And that's what card companies do. Then it's up to you to call some number and say yes, approved.
But that step is worth it. Okay. And number two steal your card. That's so cool.
So therefore you have like all these companies actually that, you know, provide easy environmental sciences research

institute.
They have a program called GIS.
I basically mentioned this, you know, and then we should move on to the topics because our Argus is the world's best location based software.
So cool. See this? Uh, not even RJ's online or just any industry in the world can use location services for searching.
Uh. I just want to go to SVR.
I want to go to Wycombe, actually. That's all. Not always online.
Okay. Finally. Here. I just want you to see, like the industries.
Oh, look at that. All right. Every one of them can do.
Search. Look. Wow. Everything in the world.
Everything, really? All of it. Profit. Non profit.
Education. Energy. You know. And you can search for any of them.
The city of L.A. knows the location of every parking meter of every park or every watering station or every bathroom.
Public bathroom. They know it all. So there is somebody searching through all of them. Okay.
That's a little search engine customized for just searching through one kind of data.
I mean, there's like, a lot of you won't go through all of them. There's even light poles, you know, like exactly where the poles are.
The city actually has a map of it. And then some maintenance work engineers actually searching for them.
So think of search as this great, you know, enabler.
Otherwise people cannot work. You know each other. It's not you doing Google search.
There's more than that. All right. So then location based search can be optimized by indexes.
And that is one of these articles talk about. Right.
And then this one is the weird thing where, you know, the bad guys can actually use the fact that, you know,
your location might be like, broadcast to other people to actually track you, or your location is never precise.
All the way to locations are highly imprecise. Okay.
But with a bunch of imprecise data, you can piece them all together and within a few blocks, guess where you live and then stakeout.
Basically hang out and watch your comings and goings. You know, if somebody is patient, they can find you.
That's so scary because you're trying to block your, you know, scramble your identity, but still they can find that.
Okay. So we'll come back to search today after I finish these slides. Similarity search is a very, very big deal.
I'm going to call this a BFD okay. It is a big freaking deal because.
That is similarity search right here. You know, the similarity search is not the same as basic keyword search is not the same
as SQL search where you type words like select something from something where,
you know salary between 10 and $20,000. That is all actual keywords going in tables.
You know here there are no keywords, no tables, nothing. That's raw data, relational data.
It went to embedding, but it still is raw data.
So if you can search through your data and get results, that is the most flexible form of search that there is.
So these new kinds of databases called vector databases, they are the ones that make it possible.
They're called vector databases because this is what they do. They provide a way to take any input from anything like an embedding layer,
and then take the output that comes out the vector and then help you store that vector.
That's why it's called vector database.
So they help you store these kinds of vectors and also index them, index them so that any vector can very quickly know what its nearest neighbors are.
The one thing that is not entirely possible. Suppose there is no indexing.
It means suppose there was a vector that came here. That's what you are trying to search for, the point that you are trying to search for them.
So you have 100 billion of these and you say, what is my top ten nearest points to my query point?
Um, if you don't have indexing, you would then need to take the query point, the tip of that,
and do a Euclidean distance calculation to maybe 100 billion other points that are in your database, and then sort them and find the top ten.
That is completely unimaginable. It may take hours and hours, right? So no such will work, but it seems to work in

seconds.
How? Because they take all those multiple piece of data, meaning all those points in 2D, for example, all those points and they index them.
How do you index it? We can take all your points in 2D and divide them into four quadrants and then say in that quadrant,
make a tree with four nodes in it and say in that node or all these points in this second node is all these points stored in a footnote.
If your query point enters here, then that's my root of the entire tree.
My tree has four nodes so far four nodes.
And each node has like all the points are in here domain here, all the points in here or in here, all the third points to go here.
And all the four points are going here.
So then when you take your query point, you take a query point and ask this bounding box, electrical bounding box.
This is my query point here or here or here. Here it can only be in one right.
The boundary edge cases. You know it doesn't have to be on the edge in general.
They can be in one area. So in our example it is on the first square.
So in this first square so then you go here. That means you don't have to look through all this data at all.
It means three quarters of the data is already discarded. That is so efficient.
So then here I could divide this into four more regions, divided into four more regions,
and say within each one of the sub quadrants what data points exist.
So those data points, they are right there in the first sub quadrant. Second sub quadrant third sub coordinate for supporter.
So then now that you went from here to here, you asked in the in that first square which are the four squares.
I mean really I mean the first one, I mean the first one.
So now I don't have to look through all these, just throw it away and just search there.
That might only have 20 points or something, right? I can easily find 20 nearest neighbors and find the top ten.
Otherwise you rapidly tree search. Why is that possible?
Because you took the time to index it. Okay. If you can index in two dimensions, you can index in three dimensions or 50,000 dimensions.
So that is where the libraries come in.
Like hierarchical, navigable small worlds, hierarchical navigable small worlds, and then even a library like size.
You know, it's a Facebook approximate search. So all of these search libraries do something very similar.
They are all index vectors. So that similarity search can happen very fast.
That's pretty cool right? Just like in databases we can search through any column rapidly, like all the students between 3.5 and 3.8 GPA.
You don't look through every student and ask what a GPA statement. Instead,
you presort the GPA column and then rapidly do a binary search and find 3.5
and find 3.6 and blindly return all the student that is in between the match,
because you already sorted that column, okay. It's called indexing to index vectors here.
And this a great thing you can index. You can do similarity after indexing based on distance like I told you, or even angle similarity.
There's all kinds of similarities okay. So this one is from the people that basically do the work.
Pine Cornell. What does similarity search say like here.
Right. Like I told you, you can take a picture of someone shoe and imagine then buying the tree.
Amazon. That's pretty cool. Okay. Water representation, say embeddings again.
You take some kind of a neural network and take words and turn them into vectors.
Okay, so this is old, but then the more modern ones, like what transformers would use,
are just glorified updated versions of the same thing one way or the other.
Vectorize it like that could be an image. That could be an image. Okay. Then you start to talk about similarity.
That means how close up, how far apart are there. And you can use all kinds of distance measures.
I told you you can do actual Euclidean distance, or you can do distances along x and y, if that makes more sense to you.
If your data is more like x, y based data, you shouldn't find Euclidean distance.
You find distance along x and y. In this case, the distance is going to be two units, not 1.44 units, because no, as the

crow flies okay.
Okay. Or you can find angle similarity. Told you there's all kinds of similarity measures okay.
There's many more. All right. So then that is all.
This is what I've been drawing on the board when you query is just also embedded in the same space as the existing vectors.
Then the search is just simply give me other vectors. The red dots are already match.
So how many reds are near my blue?
It's an easy answer, but not all data, even with this kind of embedding, would all be easily linearly representable.
Meaning there might be some data for which the scanned Euclidean distance nearest neighbor literally might not work.
You need more complex representation. That's why I talked about a sheet of paper like this.
You know, so imagine you have, uh, some kind of a multidimensional space we cannot imagine.
But then in 3D, it looks like this, you know, some twisted multidimensional space.
So in that space, a vector here at the top and the vector here at the bottom might actually be close because they might not be Euclidean close.
There might be embedded distance close. You know, for example the distance between like this and this physically is closer.
But maybe what if you search along this? What if you can search along the main.
It's called manifold. Or if you stretch along the manifold and then find closest points as opposed to actual through the air search.
If you search through the air, you're going back to Euclidean distance. But manifold search means along the actual shape of the data.
So that is the way to generalize it. We call it geometric deep learning.
We call it graph neural networks. I call it many things. But then those are all about like this.
You search for the words non manifold search okay. Non manifold search.
Some picture like what I try to show you where the paper would actually come up. We can look.
It means we have ways to, you know, generalize standard Euclidean search.
I mean, stuff like this, even C, even with this curvature is not standard flat space.
So then on that surface, if you measure distance between this and this, it is not the same as measuring a straight line.
There's some clarity involved. I mean there's all kinds of manifold like this, right?
So imagine a distance between something on this green. In other words, you want only green on green to match, not green on blue.
Even though right where the mouse is there's a green point. There's a blue point.
And yes, the closer. But you would call that far because it is not.
As the crow flies, it starts from green. The blue is not easily connected, okay.
And it's called manifold or even here okay. So again that might be a good example.
Maybe all of these are close to each other. All of these are close to each other.
But some cross points like maybe a point to point here,
even though there are like geometrically Euclidean close are not close because manifold wise they're not close.
So manifolds are fascinating. We learn a lot more about manifolds okay.
Okay. But that is where this all being extended. So then that means your search can become more powerful.
Same molecule. So all kinds of atoms. Right.
So you cannot randomly do extended Euclidean distance between atoms and say this molecule is same as that molecule.
You need to take the molecule structure into account. Like how is one drug similar to another drug.
You need to actually know the drug structure. You can just simply do like, you know, chemistry using like Euclidean distances.
The chemist would laugh at you. So then how do you know the molecule structure?
That is where the geometric deep learning comes in. So I'm going to type GDL okay.
So GDL geometric deep learning for non manifold.
Such is true geometry. Yeah.
So the first example that comes up is actual molecules is very interesting.
So in molecules you know there might be some other molecule that is atomically similar.
Meaning there's also an orange ball is a black ball. But then maybe these don't exist.

That kind of molecule is similar to this molecule as opposed to some random distance between like the two white grains or something.

So it's not about actual distance okay. It's about the structure.

So the way you do the manifold search is you turn the actual molecules into a structure, you turn that into a graph.

Then you do graph search. That is why it's also called graph neural networks, because the structure can be in general represented by graph.

Any any manifold that is not Euclidean can be transformed.

Here's one more example into a graph. Then you can do graph search on it.

So then we call it graph neural networks GNN graph neural networks.

Highly related. Not 100% the same, but, you know, similar. Okay.

Yeah. So then here we can train networks entirely for learning about graphs.

In other words, what graph is similar to what graph. So keep punishing it till it actually learns okay.

Then it can classify new graphs. So very fascinating this notion of growth an example.

Okay so Nvidia of all people made one. Nvidia has so much to gain by this.

Once again look the standard example they all use is bioinformatics molecules okay.

So what can genes do. Recent paper you know. Yeah.

So it's all about you know even neurons in the brain there's different kinds of neurons are all connected.

They're all similar. This you know hippocampus is like in a corpus callosum.

So all neurons generically are not the same. So just because they're physically close, it doesn't mean they do the same thing.

One might be for memory. One might be for music. One might be the colors.

Okay. So we should then search based on what kind of neuron it is.

Similarity. Anyway this is fascinating. You can read okay. All right.

So I want to come back. Therefore you understand a lot about vectors okay.

So vector similarity. Therefore what they showed you in this is simply Euclidean similarity.

But that takes us a long, long, long way. I think last time I wrote you here I'm going to do it again.

Maybe. What are some other vector databases okay. Oh by the way, for your homework I will use some.

Pretty cool. Um, you, one of them will use is called Melvin Smith.

Was actually in the homework in here. One of those I think the first one we use a database called Melvin's.

It's one of the vector databases, but there's others. There's quadrant is also middleware.

There's also pinecone and there's chroma and a lot more a lot more.

But the standard relational database companies like Postgres, Oracle, Redis, they've also added vector support.

So the old table databases can now do the same vectors.

And how do they store the vectors if I use tables for them okay. But we don't care about them.

But it also indexes it pretty fast you know. So we have a big choice of vector databases which is great.

And also there's new chips actually. So the chip called grok grok this chip.

Not the thing that Elon Musk make noise about. Okay, but this one is a great ship.

So these kinds of ships are actually built directly to that grok accelerator to, uh, do calculations on these kinds of vectors really, really fast.

So you can actually hardware accelerate this. I mean, already indexing is pretty amazing, but imagine hardware accelerating all these indexes.

Okay. Then even faster. Wow. So that is what this whole grok ship is like.

Was Nvidia not to be outdone? By the way, this an ex Google engineer who worked on Google's TPUs and quick Google and said I can do better than that.

So then his design grok. But Nvidia wants to also make grok like in a vector database ship.

Okay. It's amazing. Right now the chips that they have look like standard course.

They use graphics matrix calculations, but these are more like distance and embedding calculations.

So maybe you need new kind of chips. Okay. This is all very cool though.

Uh, before this though, I actually forgot something. You know, in in here.

I told you this, this one is just a funny little thing, right? Look at this. This, by the way, is Microsoft's Clippy.

So Microsoft Clippy was an old eye from the 90s, which was universally hated.

And so Microsoft retired it. So painful. You are supposed to help you with, like, word documents and finding expression on your windows hardware.

Okay, it never worked. It was more annoying than anything. So we all, like, turn it off.
The first thing you do is turn Clippy off. But Clippy came back. But I guess this one is, uh.
Yeah. So large language model trying to be funny. Okay, so that could happen, but look at this.
Oh my God, this one is truly mind blowing because this one was the year 2012, not that long ago.
In the real world of things, AlexNet, that is one of the neural networks deep learning architecture.
So its calculating speed was 1000 flops.
Yeah, 1000 petaflops. Okay. One time today in 2024, we have Gemini at the very top of this whole, almost like this linear thing, right?
This, this increase. But it's not linear. It's a log scale.
Because every time you go up one unit adding one more zero, so 1000 becomes 10,000,
100,000, 1 million, 10 million, 100 million, 1 billion, 10 billion.
So I'm going to count okay. How many zeros from here. 1020304567 zeros.
10 million times faster. 10 million times.
And then why 10 million? Add another like half, basically, you know, 50 million times faster, 50 million times faster.
Imagine that. You get 50 billion divided by 1000 is million, 50 million times, oh my God.
Right. And these models are getting bigger and bigger and bigger.
And now that's what all of these these are the very top, you know things like cloud Gemini you know llama all of these have low uh count.
These are parameters like how many connections between all the different neural networks.
70 billion. And, you know, just like 1 trillion to be 84.
But llama also comes with a 5 billion version just so tiny.
Likewise, Gemini has something called Gamma Sword Gamma to Gamma.
All small language models. Okay. In fact, one of the homeworks will do is you will actually download one of the small language models
only about eight gig and put it on your hard drive and then run the query against that.
So they can be small. They can be very large, but this is truly mind blowing, that it can be that big in, you know, that short over time.
So then imagine another ten years. I don't want to think about how amazing and how big that would be, but big is not always better.
Next week, I'll tell you, there's a whole world of people that are saying this is the wrong thing to do.
You don't just scale it up and make it bigger and bigger, among other things. This this energy problems.
You need so much electricity, but only a few companies, you and a few countries actually can afford to train like something that big.
It suddenly makes the gaps between haves and have nots. Then it will be a monopoly.
It'll be OpenAI, Google controlling everything. Okay. And Microsoft. Okay.
So all that we should leave for another time. But it's great to write, to tell you that there's so much, uh, you know, activity going on in here.
Okay. So now we understand this notion of vector d based vector d, which can be used for so many things.
In fact this dovetails with the last slide. So what should I do here.
If you have a large language model, which is ultimately a bunch of a stack of neural networks, you know, that's all it is.
It's been trained on something very general, which is all of English are opening.
I could find it. This notion of rag vector debits.
All of that. So where does the vector w1 fit in. There is no vector B here.
So the vector w was over here separately where all those things are high quality.
In this case image as I told you.
But they could also be high quality sentences, paragraphs from, um, this big handbook that the user wants to search through.
So there is a product catalog or to the particular. The large language model knows nothing about any product.
It's not specific enough, right? But this one can be all the products in the product catalog.
It might be 1 million products, all kinds of plastic connectors, you know. Or it can be some high quality textbook on physics or even C plus.
Plus and all the sentences they're all got embedded. All the polymorphism one here, all the classes when here, the recursion right here.

So you want to ask questions okay. And the question is required. So it doesn't matter what the use cases.
But you know that's what the vectors do. But those vectors are separate from the actual large language model.
But you want a large language model because Jupiter is here and you are talking to it.
You are not directly talking to the vector database at all.
Your chat still ends up at LM, but you want to somehow to actually not use its own knowledge, the training that it has, but instead send your query over here and come back with good results,
and then turn that into natural language text and then send it back to the user.
That is what you're looking for, okay. That is why it's called retrieval. Augmentation.
You augment the retrieval. Retrieval means get knowledge from that.
But in this case the knowledge M is almost nothing. You retrieve it all the augmented all with real world knowledge.
So that's one way to do it. That is called rag retrieval.
Augmentation. But there's a second way to make it smart, which is you can train.
You can train additionally using some additional data, custom data like a product catalog,
you know, or in this case the book on some layers of the existing neural network.
You know, one of them is called Laura. Next time I'm going to ask you what is the rank of a matrix?
Okay. So you can come back with some answers and ask you things like what is the norm of a matrix?
What is a rank? Okay. A what is an eigenvector?
I just talk about matrices in general, but it's a low rank adapter for Laura.
So Laura is a very interesting technique. It's called fine tuning.
So when you fine tune you actually modify the neural network itself, but not all of it.
You don't train the entire, you know, multi terabytes better, but you only like retrain just a part of it.
Enough for it, enough for it to answer your question okay. So then that is called uh fine tuning.
And this one is called retrieval. Augmentation. There are in general two very different techniques actually.
You know, in the real world you can write applications actually that can combine them.
The reason why you cannot do fine tuning most of the time is because you and I don't have access to GPT four.
OpenAI is not going to open up the whole, you know, the gold mine and say you train it on whatever you want right now.
This is an API that controls all of that, right?
So we have no access to the actual real alarm, but there are many open source alarms you can download where you have 100% access to all the layers.
So if that is true, then maybe you can do fine tuning. Mari's better than rag.
Okay. There's all kinds of things we can do. Order both. But now why pick one, right?
So then that's where all of this was at. The whole generative. Okay.
Let's call infinite memory. When you have external memory like that, this memory like that.
Right. We call this infinite memory. Because compared to what it knows, there's no limit to how big this can be.
You can even, by the way, have multiple vector databases. Okay. You can have multiple vector databases.
And the query can go to all of them. And you want to combine the results of all of them and present that back to the user.
It's truly many fun things we can do. That is actually what agent programing is.
So you can do agents session just means one from that does one thing.
So maybe I can have one problem. But the problem breaks up into three problems because we know there are three vector databases.
And each problem goes to one vector database and retrieves whatever it does. And then the problems can even faster access to each other.
And finally it will all come back to you as some amazing answer that you, you know, couldn't have looked up yourself.
So very cool, right? So in programing. But that is why the notion of infinite memory comes from.
Okay. So let's see what else we have. Yeah.
So this notion of, you know, new world of virtual augmented ML, you know, this was written last year but still highly relevant.
You can read. Mhm. Okay. You know because it's Q and I might as well watch some of it.
There's got to be trainable enough. Now that data retrieved is obviously a few items.
So we see the prompt. That's what the prompt looks like. Answer the question based on the context below.
Okay. And then that will all get filled with actual result template based programing.
So that will get replaced by an actual question from the user. And this will get replaced by an actual answer that NLM

comes up with okay.
We call this a prompt programing language. Lang chain is one of them but it's all llama.
Llama index action is under the one llama index. Mention that they have to make your prompt in that format a programing language.
Now and then, because this generation model is basically just taking over the text that we have so far and continue on, it was long term encoded by GPT cc model.
That is the external memory within pinecone. So we essentially have a ton of index items like this.
Right. And then we introduce this query vector into here.
So maybe it comes here and we're going to say just return.
Let's say in this example we have a few more points around here. Just return the top three items.
So in this case we're going to top case.
If that is your query then your query is going to match like that that and that point maybe but not this other point okay.
This. And then you can rank them.
So the top even the top three can be ranked in some way, not just based on distance between what it's called reranking.
And the first rank is based on distance. But you can change the order here at the last generation model.
And here these data points we actually. Right.
So the cool thing is those that come from the database, from the external vector database,
might be actual chunks of the answer that you want the full not English even.
But their job is to combine all of them and then present that cohesively.
That's pretty neat actually. Okay.
You could do language generation, but the generation is happening from the vector retrieved that is required to translate them back into okay.
So we can keep going giving. And then you see that vector is actually what is what is actually not true.
Okay. So then look at that. You can actually ask. That's pretty cool. We do that here.
Well and you can see like thread tags and this question asked.
So you ask a question and it's going to answer the question. And then good enough for you right.
So it's very neat. You know for instance this one that. So who was a 12th person on the moon.
Okay. So this one might not be common knowledge at all. Most of us do not know.
Right. So then treasure breeding might not know but say there was a NASA database.
So NASA database obsessively maintains records for every single mission ever happened throughout NASA's history.
Then that will be the external augmentation.
And then that question can mention you can even answer what are what are all the people who ever went to the moon and solve them for you?
So you need that good knowledge, okay. By the way, here's the ironical part.
Okay. So knowing all of this, you need external memory. But about a year ago when Bart Bart came out.
So initially Chhatrapati came out, Google was basically and all external, you know, caught and said, oh my God, we need to do something.
So quickly the road, Bart and then made a demo which is an animated GIF, you know, plays over and over,
went to Paris for no reason, and try to preview the amazing Bart all over the world.
And so this jiff, you know, the stupid looping. One of the questions in that thing was you asked Bart, um, when was the first exoplanet discovered?
Simple question. Okay, okay. And then the answer that Bart came up with was a few years ago, using the James Webb telescope by 2022 or something.
But this is completely wrong because the first exoplanet was discovered in the 1990s.
Okay. So more than a decade ago. So the very first out of the gate, Google's Bart, which is the answer, is pretty screwed up.
So that night, Google's company valuation fell by $100 billion.
So you should Google that and find out for yourself. Okay. One night, the company's value went down by $100 billion because there was a big screw up.
Okay, some human could have checked to see if that's the right answer or not is so embarrassing.
But retracting. At the time there was no single rank, but rank would have actually learned that rag would have given them the right answer.
Okay, that is how powerful, I guess. But Raja's problems also, which is, uh, when you take your document, maybe a

PDF file,
and then you have to do this thing called chunking, some of your homework will go through all this.
You can read chunking means I have to break up my big PDF file, or my CSV file, or my text file, log file, whatever, into little pieces.
They're called chunks, and each chunk becomes one vector that you embed.
You don't embed like a whole PDF file into one point, right?
I mean, you can mention you cannot take all the words in a book and make some crazy vector of all the words.
You got to break it up.
So that creates a problem then, because the answer the user is looking for might be in one of the chunks, and the other answer might.
The answer might continue in a second chunk. But somebody has to piece them together.
They might they might not know how to properly piece them together.
Okay. So you're basically stuck at the boundary. The answer is not fully here or here.
It's some intelligent combination which doesn't know how to do.
But thankfully we have new ways to solve that as well. So there are all kinds of problems, but quickly people come up with solutions.
You know, we want to work with completely want rag. Rag is a great thing.
We. I make too much fun of all kinds of I, including all of these.
But, you know, Reagan willing to admit it's a good idea. Okay. Okay, so let's move on.
I want to tell you a new thing for today. Yeah. So this notion of latent Dirichlet allocation called LDA.
This is usually considered a data mining technique. You know, that's actually what it is.
But the whole latent why is it called latent? Because these kinds of spaces with 3 million dimensions that don't look like English words are pixels.
We call them a latent space. We call them hidden space because that does not look like a word or a song or video or something.
It is purely a mathematical dimension with some access. You not, that's all. Well, latent space.
So in the latent space you can cluster Dirichlet allocation.
Dirichlet means Voronoi polygons.
So then if I have two dimensional latent space like this and then I have a cluster of documents physics here, chemistry here, biology here.
That boundary would be physical biology. This boundary would be physical chemistry.
This boundary is going to be biochemistry. So therefore by simply making these virtual polygons around all these documents, I can cluster them okay.
In some way. That is all LDA ultimately boils down to. So the name is pretty cool latent replication.
But if you Google this you might see an image hopefully just like what I'm showing you.
Let's find out. Okay. Yeah I see like here.
You know that could be one topic, second topic, third topic. And that's the boundary between topics here.
There are not using polygons. But you know, I told you that like bubbles are muffins, when the muffins start to grow towards each other,
well, then intersect and stop running into each other and that will become a flat polygon.
The then they are flat. Just so you know, that is what a Dirichlet polygon anyway is.
Okay, so you can see many examples, but same idea,
but in the end is about topics and how you can classify a new topic, meaning a new book or document is given to you.
What category already assigned to. Yeah. That's a great question.
You can. You absolutely can.
The whole Dewey Decimal System is incredible because long ranked member Dewey said, for physics, we need to have a number like 600 or something.
But then 610 would be mathematical physics. 620 rhetorical physics.
By using decimal points, there's no end to how many decimals, right?
You can have knowledge with the knowledge and knowledge. Yeah. Likewise you can absolutely cluster.
That's actually very cool. It might be called multi resolution all.
Let's find out okay. Multi resolution Dirichlet.
Uh, allocation. Or hierarchical.

There might be hierarchical. Dirichlet. Uh, I'm going to say hierarchical.
In fact, I'm going to say a hierarchical LDA. If this does not work, okay.
Yeah. So right there you can definitely make hierarchies okay.
It's a good idea because in the world so many things are hierarchies anyway.
So you can have definitely topics within topics within topics. Sure.
Yes. Cool. Okay.
Good. What else? Um. Okay, so that's the LDA paper that I clicked on.
Okay. This one is simply a library. Uh, there are many libraries, but faces a Python library.
You can just pip install and start using it. You know, in face, you can even give it an English sentence and tell ask face. Show me what the vector looks like,
and then you can give it a second sentence that looks like a lot like the first sentence and ask, what does a vector look like?
You can see that the vectors are pretty similar and you can ask do the cosine similarity?
Do a dot product between the two vectors because the numbers are all the same, right?
So the dot product the angle is zero close to one. You can even calculate it yourself.
So face is a very friendly library. You don't need to do a whole lot, just play with face all by itself okay?
If it is useful and face is used in so many applications obviously.
And it has approximate neighbor. Oh okay.
Should I say continuous then. No idea what that means. Uh oh.
You know, if all this doesn't work, it doesn't matter. I'll show you a first tutorial.
Okay. Check this out. Look at this. Literally a face python tutorial.
All right, Pine County. Okay, good. So what makes it good?
You know, check out the video. Okay, why don't we just do a little bit here so you will know what fresh.
Hi. There is a multi-dimensional game. Easy. We're going to be covering the same guy.
Similarity search all things. You must obviously be one of the prime question covering what science is.
If you drive to San Francisco, Pine Cone is actually so close to opening AI,
using it and in the same area introduce a few of the key indexes that we can use.
This is called the missing manual. See that? So just as a we partitioning the index that's the whole point right.
Indexing as you can the name it's a similarity search.
And it's it's a library that we can use from Facebook I.
Okay. So I'm going to keep going to and let's say here this is our query vector.
Again it's always about queries and finding nearest neighbors okay.
If we were comparing in thousands of dimensions not just something like the distance between every single item.
So that's what I told you. So when you have 100 billion items, you cannot do similarity to a closeness match through all of them.
It's always so much time. It's completely pointless. Find the vectors which are closest to it.
So you need to somehow partition. This can optimize this.
We can even quicker load of sentences in a notebook.
You can play with this. Pretty simple. And then what's written in as a normal file.
And we just write lines equals FP to read literally like in front of you.
And we write indexes in notebook you can play with. Okay. So I'll go through now how do we add our vectors our standard embeddings.
I uh like that. So then see that work.
This what I told you right there. You can give it a sentence and say encode.
And you can actually see the raw numbers that the sentence turns into, like right there it is.
So if I then change this to someone runs with the football, then maybe some number here might change a tiny bit.
So you can see that sentences that sound similar in English would have a similar representation in the embedding space.
You can literally see it okay. Wow. And then no wonder the dot product is going to be the same across these four items.
Now these align to uh lines.
So the the text that we have appear that will align.
So what we can do in this case by the word watch will be different from what I said, but still some idea it is one of the results show is retrieved.

So your search already had some numbers. The retrieved value has numbers.
Also the numbers look pretty similar. I call but look how easy it is so you can actually play with phaser.
So that's a technique to very see that you see when your sentences are very similar,
then the embeddings will also be very similar or similar words are.
No wonder such will work. This is one of those things where it's guaranteed to work right there.
Now we want to write here which is a query vector.
And let's say again it's all about similarity, right.
The closeness, the Voronoi, which is published over and over and over again in this course.
You've seen it so many times. So we could have, you know, we could have millions in each.
So, um. So there's a lot in there.
And if we were to compare that query vector in this, this thing here to goes right back to terms frequency.
You know, tf IDF it's the same idea as to take loans decades old.
This idea in some sense vector search is not new at all. What?
This approach allows us to do is incentive goes back to 1970s vectors.
Just check it again. 50 years old. And once we figure out, okay, I won't bore you with all the rest of this endless parameter.
And then from now we can.
So our quantizer, which is, is almost is like a because I call you know, he's doing live coding and explaining what is actually happening.
If I were you and I would watch all of this and fully understand clustering now and you're doing pretty well.
Okay. What we do is we we end up getting close calls and all I typed was for this tutorial.
But you will have many, many, many more. So what about just one more as an example here.
See again there's the embedding vector. You know directory looks like that.
And Euclidean distance you can say this multi dimensional and just call it you know I you a number of dimensions 1 to 3.
And then your prototypes did pip install five CPU. It is so easy.
No need for GPU also. And that is the embedding layer.
You can use a sentence transformer and then it will take any English words like that and then turn that into an embedding.
Okay okay this is the key point. Create vectors from text.
The sentence transform is used. So then every piece of text will become a vector.
You know that. Then you can do a similarity search on it. See like right there.
This is the whole cool part right okay.
And then you index all this vector is all the same steps over and over.
And again your query also becomes a vector.
And your query would then become localized, meaning it will find the points already in the database close to your query vector, you know,
and then you find the top nearest link right there and find the top k in our case like one, two, three and then sort them some kind of ranking.
You can sort by distance. That is a default ranking. But a new I some of the I cohere.
So cohere is a company also in the area. They have a Reranking API.
So what they'll do is take these k ranks, maybe three ranks.
You have to give it to the user as ABC. But then and will look at that and see, you know C is a better answer.
We should do see first and then and then B it'll change the order that the user would like to see it and it's a better order.
Wow. So you can then play with that library Coheres Rank library.
And you will compare the before and after like, you know, see it without the reranking seed width and you will always agree the reranking is better.
It never makes it worse, by the way. Okay, uh, in the worst case, I'll leave it alone,
but it'll usually order it in such a way that it's a better ranking because this is simply Euclidean distance.
But then the new network will actually examine the words and then go in some kind of a concept space.
And okay, but look how cool this is, right? Like that. So then you do search based on this kind of a closest distance.
Wow. Great. So then just keep going. Anyway, look how cool this whole tutorial is, right?
Here it is. Again all the tax code encoded and they became vectors.
And your search is also a vector. And face will do distance calculation and then give you the results in the meanwhile.

How does it do a fast search. Because when you basically fully understand this little architecture diagram right.
Somebody wrote. All right. Us now.
So there's so many you guys just. It's all I would do if I had more time.
Just keep doing them until it becomes second nature.
What else? Yeah. Okay. So the notion of task specific agenda one showed you a little bit last time.
What this is, is what if you want to I'm sorry.
Create crazy abstracts from nothing generator AI for coming up with the scientific papers.
Abstract. You haven't written the paper yet okay. But you would hope that the abstract looks so cool.
Wow. I should turn this abstract into full time, full paper. That is what this is about.
So weird idea. Because we don't write papers, though. Okay. And where does how does it even know how to make abstracts?
Because I took 100,000 actual scientific papers, abstracts and train the network on all of them and said to it per year, new abstracts.
It is literally doing word based, you know, embedding search and similarity.
In other words, it might produce completely bogus, crazy stuff because it doesn't know science.
It doesn't know anything. Okay. But the abstracts all have good words anyway, in some good order.
So the hope is at least a small fraction of them might be usable abstracts.
And then your job is to turn them into a paper these days.
By the way ML which is such a pity. You can take things like abstracts and turn them into long documents.
You can do what's called summarizing, which is the opposite. Take a whole document and turn that into one paragraph.
But we have for good or bad things that can go the other way, usually for bad.
You have no idea how many scientific papers as we speak actually already have been published by what are called predatory journals.
Who will take $500 from you and publish any paper that you provide them?
They won't proofread, they won't do anything.
But sadly, indexing engines, actual scientific document indexing engines will index those papers for good or bad.
So when a real scientist goes and searches for papers, that bogus paper that you wrote,
that you wrote actually will come up as a real, uh, you know, answer and the scientist goes looking for it.
Wow. Uh, in nature, you know, 1994, I remember reading, like, a ratio.
I don't remember this paper. Look where it's come from. And the answer is that paper does not exist.
So it finds magazine named like nature. It finds volume numbers like form.
It finds page numbers like 190 to 200 and find some author's name from somewhere and makes it an actual scientific citation.
Bogus scientists, bogus citations created by I.
So that is possible. Check this out and then we can get back to what we're doing.
Select bogus citations. Okay. Produced by ChatGPT.
These things are so real by the way. And this is a serious problem.
This is a serious problem because real scientists are misled by.
Mizzou University of Missouri. They actually have a little article about this you can read, right.
Um, it just makes make stuff up. So you need to be very careful not to trust that.
In other words, don't ask Lybrand what are the paper go, because it's a bogus paper.
It's sad. Right? Look at that. Fake references. Fake citations.
Fake court citation. This is even worse. By the way, this happened last year.
Some lawyer went to court and argued for his client, citing some previous law legal, uh, ruling.
Except that previous ruling did not exist. Temporarily.
Completely made it up. So the lawyer was fined for basically lying to the court.
Okay. What the [INAUDIBLE]? A lawyer check. Look at this one.
See? Fine for submitting fake work. I mean, this thing is going to over and over.
It's a very bad year. All three days ago.
Right there. Fake but plausible. Study science.
How many? Uh, in psychology. 6 to 60%.
Citations are false. Holy [INAUDIBLE]. All right. You want to take the middle number 30% or something?
One third of any citations you see in a psychology paper doesn't exist.

And we're just getting started with all this. So in ten years, imagine it will be so polluted, you can never know what is really right.

The papers are fake. The citations are also fake. Scientists, real scientists initially would object to all of that and go out and stop it.

But they'll get tired of calling all this B.S. out because what is coming is a crap flood.

A scientist would basically be defeated and say, screw it, I don't care anymore. And at that point, things will totally go to [INAUDIBLE].

Okay, so that is actually what's going to happen because we cannot police all of this so much.

It's not even funny. Look right there. Right? Um, again and again, referring to work that does not exist.

So anyway, this kind of a thing is a bad idea in a sense. In other words, even though they made this thing, what the heck was that?

This one. Okay. So, you know, I fine tune GPT, but that's fine tuning, by the way.

So you go in this last layer and then you teach it how to write scientific abstract again.

I mean it's a very weird experiment. It succeeded for what is what I show it to you.

It writes abstracts. Okay. Okay. Here's the Python code. You can actually run all these yourself.

GPT two is free, by the way. So if you wonder why they used to call GPT, it four is not three.

And three is pretty big. Two is small enough and free that you can run it in on your own machine.

And two is still highly worth experimenting with. One two came out.

OpenAI made such a big stink when 1.5 came out, OpenAI told the world, wow, we have this new dangerous er that can change society.

We can even release the whole model in one piece. Well, at least the main chunk.

I mean, you know, this all basically in retrospect, but they made a big deal out of it.

So if that is true, that two is better than 1.5, right? Okay.

So you see load data set is simply that's a piece of text okay.

That somebody had split equal to train two training data. Test data.

And on the average you know each abstract had like so many words okay. Doing some little video on it.

But yeah just go and actually train it tokenized data so I can take all the words and tokenize it like create embeddings and then fine tune it okay.

So it means change the last layer. So the neural train validation.

Okay. So after you do all of that do n compile fit evaluate the model.

Okay. And then what happens. That is standard machine learning loss like cross-entropy loss binary loss.

You what. All right. And then what does it do in the end.

Uh, perplexity language model okay.

Hopefully they show you results. You know, that you can actually see here. Yeah.

Right there. Output generated by the model. Check this out. Oops.

See that? It's weird, right?

Whoa! Okay. Mhm. Yeah, I made, like, an image out of it.

Just, you know, like, right there. So this all entirely written by.

I said, somebody wants to write a paper about recommendation engines.

And the idea is it starts off with this weird abstract. And then hopefully, if you think the abstract is okay, you can turn that into a real paper.

It's not bad English wise. The grammar is not bad.

But that's not to say anything about the veracity. The truth of what?

[INAUDIBLE], whatever that is. Okay? I would personally never use it as like a weird idea,

but it definitely works because if you google any of these phrases, Google does not come up with any results.

So it's not plagiarizing existing, uh, abstracts.

Why? Because all the abstracts went in this embedded space.

This space, by the way, is very cool. So latent space, right.

This space of representation is a very twisted version of music, video, images, text, raw sentences from our training data are not sitting there.

Raw pixels are not sitting there, but turning it into something so alien.

The representation, whatever it produces, is also alien in the sense it is not plagiarizing from anything.

That is why the art also that it generates is very beautiful,

because it's not really cutting pieces of my eyes and your nose, it's actually generating it.
Okay, so something could be said about that, but at the same time, it might not be factually true, it might not be technically accurate okay.
But it in papers anyway. So that is one more thing I wanted to show you.
I want to move on and maybe finish this up. Okay. And take attendance and take a little break given.
Go ahead. So we are almost at the end of this, and I am saving the rest for the second half, which is we're almost there.
Yeah. So then there is a notion of fine tuning as opposed to external rag.
You can do rag on this also. Okay. But this one is fine tuned with fine tuning because I had access to it.
Okay. What else? Our last slide from last time.
Yeah. Again same idea. So this is again all about this notion of, you know, embedding, um, you can embed,
you know, different tasks like in different layers or even different networks.
Okay. The way that I'm going to summarize this whole task embedding is this why have just only one LLM?
I have one LLM, right. And then one rag, you know like external database.
But instead if you even have multiple even multiple elements and then multiple external memories,
then what we can do with all this is external memory.
You can write some program with a bunch of prompts.
That program can then go in this first alarm, use this external database and come back with an answer.
Right. And then send a prompt vessel alarm for the database and come back with an answer and do it a third time,
and then chain all the answers together, and even send that to a fourth LLM and summarize it all and come back and give me a result.
At this point, this is now looking like a programing language.
I'm making function calls and I'm aggregating all the function calls and making one last brand new function call.
We are doing the orchestration. That's collision programing. So you just call an agent.
So we do the agent programing and so we know what we want. And therefore it's very smart.
Oh by the way this also something crazy. Uh two weeks ago I was here and I brought up Devon and I said, Devon is the world's first engineer.
Software engineer. Then last week, I believe I brought up Derica so that, you know, it turns out that Devin was fake news.
Oh, my. Holy crap. That guy, okay, that sat there in confidence and look at the world for let's be a scrub because he used up work, okay.
And there was basically monkeys behind the scenes typing okay really fast. The whole thing was a staged demo.
The guy got millions of dollars, right? But it's so shameful.
Okay, it's horrible because in the meanwhile you guys think, oh my God, there goes my job.
Okay. But then, uh, shocking.
Uh oh. Oh. I want to play this for you for a little bit.
This is the internet of bugs. My name is Carl. And that.
There's a lie. So this video is in three parts.
First we're going to talk about the claim. Hey that looks like my office.
We're going to talk about what should have been. Hello. What Devon actually did and how it did it and how well it did it.
It is shameful. Okay. I mean, the guy had no business lying to the world.
I am not anti. Yeah, it's gross, but I don't know about their economy.
So my source. So hopefully when I'm doing the talks Devin was interviewed not quite a month ago now.
His character was because, unlike the, uh, engineer and copilot that writes code,
the documentation just publishes a quote on the website that I'll put in the description.
So that's multiple. But the whole thing with the line, it's the first line of the video description.
Okay. So watch dev and make money taking like did a lot more planning why this is a fake cases or I fake scientific papers.
And then there's the prominent ones. And this hurts real software professionals too,
because there are going to be folks that are going to trust the code that AI generated through just means more bugs on the internet.
And they're already way too many already. They're also purposely made the in write bad code and actually purposely pretend to fix it on my car.
It was lying on top of a lie. Okay. Okay, so the claim on the cover is very beautiful.

You know, um, another video that I subtitled, so please watch it.
The job of a software. Exactly. I mean, that's what scares you guys.
I'm not getting a job. I don't have the important part, at least for now.
Is that part of being a software engineer? Is communication with the customer with cost is going to go up because you want to bid as low as you can,
but you want to make sure that the customer understood. Anyway, it's a pretty cool experience.
Sample data is fine. It turns out this all this scripted demo is what it was.
Normally it should be more complicated than that, but that's what changed the way I did that.
Any idea how? So I'm glad this guy did this video, but it's pretty shocking in Reddit, right?
Absolutely. All right, let's just blast it. I'll see you later.
Yeah. All right. Okay. Oh, yeah.
Some YouTubers have their content. Okay.
So, uh, one more window. Wow. 14 hours ago.
Oh my God. So, at least for now, the world is safe.
It's 628, so I'm going to call it attendance. Okay. Uh, Deloitte is having a presentation.
About how to be a data analyst for Deloitte. I think it starts at 7 p.m.
I don't know who's planning to go, but at least as of know you guys didn't go.
I appreciate it because what Sal, who was actually in my course, he's the president of Grids.
I'm the faculty advisor for grades, by the way. So I have the split dilemma.
Okay. So, you know, should I let you guys call if somebody wants to go?
By the way, you can actually go, okay, I'm not going to stop you. Yeah. So that's you.
Can he put it up on Piazza? You can watch this on video if you go cool.
But meanwhile, since you're here, might as well do this, right. Oh.
Uh. Cihan. Lu. Cihan. Are you here?
Oh, wow. I saw you, Cihan. Okay.
Z1. If you weren't, you might be in the overflow section.
By the way, the overflow room is almost as big as this.
This class is about 450 students or something. Going gets crazy.
Hey, this fall I am not teaching 585 for the first time in 14 years and only one time I'll be back to it in spring,
but I'm teaching 572 this course, so it will be a massive, of course, probably about 900 people.
It might be involved, actually. Okay, so who is John?
Genuine. Oh right there. Okay. Thank you. Cool. Uh.
Javon tour. Hey. That's Javon. I know Javon.
Oh, my God, all the way in the back yard. I saw him twice last week.
Go out. Okay. Um. Got graphic.
Graphic, graphic. Okay. Wow.
More analog. So far, it's 100%.
Right. Sorry I can't.
Right. Sorry I can't. Yeah. Okay. I guess that right.
That's what you do with the neural networks, by the way, a mask, a part of the output that's called masking.
And you're asked to guess, and then you reward or punish the weights.
Okay? Okay. We should stop. Uh, it's 630 now, so why don't we take a ten minute break?
Okay. A nice ten minute break when we come back. We will definitely do miscellaneous part.
More new things to tell you. So please be back in ten minutes. Okay? You can talk to me, okay?
Not that. It's kind of like.
Um. I stopped by my class, like, because I like, I was, I kind of charge my computer right before my lab, which is over here.
And I noticed your voice. I was like, oh, okay, that's very cool. I say, and what do you want to say?
You did mention like, the, um, the, uh, the language, like, uh, uh, top ten, like stuff like languages of the year or something.
I forget, like, so. Yeah. Toby, it's called a should index.
I know, like StackOverflow does the same. Like when they do, uh, developer survey Omniverse or something.
They call it some of, you know, every year. Yeah, yeah, yeah, this one is more serious.

I mean, it's more comprehensive, but the GitHub one is very cool because, you know, they make a whole page out of it and it's very beautiful.
I know yeah I get up it's beautiful actually every year the team is different okay.
But yeah they all track in all languages, you know. And you look at things like that and then the top or a few.
And so what's fascinating and JavaScript is always on the top right Java.
But now rest of that becomes geography I don't know I don't fire [INAUDIBLE] YouTube account.
I'm not sure if you're familiar with that. They do. It's a you should look at some of their fire ship.
Look at it every year. He does like a summary of like all the StackOverflow.
Uh, this is very interesting.
I will definitely look at it for sure. Thank you. All right.
640. Hey, cool. We have still our part two and hour from today, so please come on by.
Uh, if someone wants to. By the way, go to Deloitte, you can actually go.
You know, I don't mind. I want to in attendance. Okay. Yeah. I'll be nice to you guys.
Yeah. All right. Check this completely out of.
Well, you know, it's somewhat related, actually. And then we'll get back. So WebAssembly sent.
In all our features okay. Even with all this, all of this, they even vector search, um, transformers, embedding files can all be done on the GPU.
Can all be done in containers. Can all be done on the cloud.
A GPU cloud really like a black cloud now and it can also be done with microservices.
A couple of weeks ago. That is what Jensen Huang announced at the Nvidia GTC conference.
They call it Nim nim Nvidia inference microservices.
So at Nim you turn all these calls like here's a query, do top five, rerank them, and then summarized and put me into one function call.
You know that Nim would give you a library and then you as a developer can do rack applications by bit ten lines of Nim calls.
You simply piece things together. Okay. And if you're asking me, where does the Nim call run?
Where does each call run? We don't know. We don't care as a developer, because each call can even run on a different GPU on a different cloud.
So multi cloud solution. It is purely magical. So your job is just to write the code.
And then somewhere somehow magically it runs and comes back with the results okay.
So I call the acronym MC. Hey you guys you can talk in the front.
Sorry. Okay. Uh, are in the back. So please listen.
Okay. MC stands for microservices, which is one function call at a time.
Function call and then return. No class hierarchy, no master.
You know, uh, architecture is pretty loose and very loosely coupled for a function for a microservice,
all you have to know really is the function's name and the function's input in terms of is that a string type?
Is it a floating point like a temperature value? Is it an integer for how many results you want and then some kind of a a type for the output?
Like what does it return? An array of strings? Okay, that is all you need to know.
It doesn't matter what programing language that microservice was written in.
And that is pretty amazing. Uh, could be written in Python. Can be rust, can be C plus, plus can be Java.
It really should not does not matter because a microservice is a microservice is a microservice.
Uh, again, like I said, if you give it a query and then some kind of, uh,
context for the query and how many results you want, that should be the answer.
End of the discussion. Right. That is what these are.
So these microservices, they go on containers. That same query, that one microservice is not just running on one server one time.
So a million people use it. They all have to queue up.
But instead that same query, that same little, uh, function is put into Docker containers and then deployed on a cloud somewhere.
And then the demand goes up. More and more people are doing such. The containers can duplicate.
Docker can be the first talk again and second third for 100 Docker containers.
And then Kubernetes is a container orchestration solution that manages containers.

When the demand goes down, many containers automatically shut down.
It goes back to the default five. It is very amazing right? When you write programs like that, there is a notion of MSK.
The reason I tell you all this is, uh, the programing language.
So now even as we speak today, we're still used to picking one programing language like maybe go,
you know, a dart or, uh, Swift or something and writing your application in terms of functions, maybe.
But still there's one architecture all in the same programing language.
But suddenly the notion of a microservice being in terms of functions frees you from using one programing language.
You can use any programing languages, you want, multiple of them. And the one practical technology that will actually make it happen.
Because the big question is if you make a Java function call, it needs to be in some weird Java container JVM somewhere.
How can I take the string array the Java returns and give it to a Python function with df definition as an input?
Because Java and Python can talk to each other, right? Python interpreter is not the same as a Java compiler, the JVM.
So we have a new solution. It is called WebAssembly Wasm.
So with WebAssembly you write code in practically one of 30 different programing languages.
Everyone, every language you ever known C plus plus, rust, Python, JavaScript.
And there's a compiler for each language. What the compiler would do, it's called a transpile or not really compiler.
It will compile no matter what language you wrote your code in. Into the same kind of a binary called WebAssembly is called a dot wasm file.
That is what these are. So dot wasm file has a name, you know, it's got a function call like all of that.
Right? But that is loosely analogous to a JVM dart class.
So a dart Java file Java compiler turns it into a dart class.
If you look in a dart class, you will actually see assembler instructions move or one compiler to low register R5 into memory and add it.
Add the number five to it. It is a virtual microprocessor. Okay, that's what the JVM is.
Likewise, WebAssembly also runs on a different virtual processor, which is basically run by every browser in the world.
So you say, now that's how Chrome runs. That's probably how this page is running right now.
But until all these years, as many as WebAssembly was tightly coupled with browsers.
But now there's a new world where anybody who is doing machine learning or.
Processing computer vision. You're saying. Wow.
I want the same flexibility to write my code in any programing language and combine, compile them all into Wasm and interoperate.
So the reason why all these name services can all be written in any language is because in the end, all become wasm and wasm is calling each other.
There is an incredible water, right? So. Oh yeah. So what do you do with Wasm?
You can put Wasm on a server and load into your browser.
And one line of JavaScript called JavaScript has a fetch API.
Just say the word fetch and give it a URL, right? The same fetch can fetch awesome from anywhere on the web and then it will run.
So that is the magic. So WebAssembly can be run on actual laptops or standard laptops or phones or browsers or servers or clients run anywhere.
But it's all the exact same data types, completely portable.
So Wasm programing languages just say Wasm languages.
Okay. And then I want to show you the little synthesizer and we can get back to our actual part.
Two of the next slides. WebAssembly compatible languages.
There's a whole list of them. Okay. Awesome.
Awesome. Link. Get ready. Okay. C c plus plus rust.
Go dot net C sharp. Uh, f sharp.
Love you. TypeScript WebAssembly.
So you can directly, by the way, a program in motion. That's a lot of fun, by the way.
You can learn the little microprocessor language and directly write coordinate okay.
Okay. Many device drivers already know that's a Google language called dart.
You see. Right. Java JavaScript. This is amazing. What you're saying is any of these languages is a new machine

learning language from MIT.
You can take all of TensorFlow or data mining algorithms and convert them all into WebAssembly and call each other interoperate magic.
So that is what this is here. Somebody used rust, which is one of the newer programing languages, and web audio.
It's a standard web interface for audio generation. So they use all this to make a sound.
Okay, so the synthesizer is entirely browser based.
This. So can programing rest and render on the browser.
My name is Casey from mosaic and this is a video that's a bit of a guide.
You can see that I have all this. That's calling a function on it, since it is a different function.
Sawtooth filtered slightly. When you have an LFO, you can start doing drums.
Okay. So if we. A frightening operator to modulate.
Fantastic to show off the filter. So what are the hackers does with the actual.
Check this checkbox. Filter will be enabled. Implementation.
Like what can run this? And it is great if you like audio generation.
You can then go on like you know watch this little video. Why the heck?
I just want to run it. That's what I want to do. And then get back, okay? Oh, my God, it's not there.
One last time. Well, some sense can.
Oh. Oh, this isn't okay. Oh, look at that.
Oh, yeah. If I have a Midi keyboard.
I should play the keyboard. I work sometimes. On screen keyboards as well.
I mean this is so great. That is Adsr. That is a reshape like any single sound.
Okay. I mean, this this all programed entirely using WebAssembly.
I mean, how crazy is that? Okay. So then that is exactly what this Y Combinator page is talking about.
Okay. So anyway, so you can go and have fun with the whole WebAssembly okay.
So you can look at Wasm basics if you want. And I want to actually show you Wasm machine learning.
Then it's more to our point. So entire search engines can be written this way.
Anything at all. It means like right there you can write TensorFlow,
dot JS or even TensorFlow in Python and convert your Python JavaScript code into
WebAssembly and then select stack based virtual machine exactly like Java.
Okay. In other words, if you bluntly want to know Java, set up Sun Microsystems Script Java and Oracle bot sun and kill that.
Basically, Java is not dead. Obviously I shouldn't say that. And it's running on servers and things, right?
But Java has no future. It's an older programing language. It is so damn slow because they want to not be like C plus plus with dangerous memory.
But then the price to paid for it is speed is horribly slow, so you cannot use it for anything.
Real time self-driving car would never be in Java. Rust is like C plus plus, but actually uses the same kind of memory allocation but safer.
So if anything, rust will entirely replace you.
Okay, but now the point is WebAssembly is here and that will not be screwed up the same way that Java at all,
because the people that made WebAssembly, they are basically a mozilla Foundation people.
They are the ones that wrote Firefox. Brendan is the guy who wrote JavaScript, who invented JavaScript.
He is part of the people behind things like WebAssembly, so it is not going anywhere, meaning it's going to have a pretty bright future.
See that? So you learn to write your anything.
Neural networks, decision trees, uh, support vector machines, okay, whatever the heck.
Really. Clustering algorithms, all of them do it straight and then one day they'll all interoperate.
How will they interoperate? Was it was is the standard interface.
See this was dev API specification for software.
Secure applications from any language can be run anywhere from browsers to cloud embedded devices.
Look at it has a pretty bright future. Okay? It means you don't have to be wedded to any one programing language in a team.
People can come in knowing, Swift, knowing Java, knowing dart. It doesn't matter.
You know your code in whatever language is comfortable to. But then let's turn all them into WebAssembly and

interoperate.
So notion of interop writing okay, that is crazy great.
See like here compose software written in. So we don't need Rest APIs or Rest GraphQL that all go to [INAUDIBLE] okay.
You don't need like any of them directly have web wasm like communicate.
See this. There's so many I mean you can just try any of these.
They all just work, you know? So cool. Server side edge devices self-driving course.
So they have a component model. That's what I mean by component model.
Component model. So like a black box actual function all you need is just simply input and output say like right here right.
Single Wasm file inputs outputs, core modules. You know core modules Wasm right.
Yeah. Agreed upon way to expose type C like right there. You know, how do you export one type to another information that how the API.
Right. So that is exactly what this is. Component model anyway.
It's really beautiful. None of this is new by the way. It's all this how you program in hopefully any language okay.
In other words, hide the internals and make everything be about functions input outputs.
And you can change them. But now suddenly they're actually across, uh, programing languages.
That's a new part. Send video name. See this production ready APIs run anywhere.
So this is where you can try all the see all these large language models.
I told you there are many links you can use for your homework okay. So Mistral is one of them.
And so then Nim can actually make you write code.
Same name can have you like you run things from any language.
In this case are running nim from Python. Okay. Python.
And you can run nim from JavaScript or even shell programing like a bash shell programing.
Whoa. Which shell variables? So Nim is pretty great.
So like right there Nvidia Nim set up easy to use microservices.
That is why this is all going. So we don't even need, you know, things like llama llama indexing or lang chain.
I mean those are great. But the Nim as observed all of them and then turn them into a translation API.
You know, maybe a question and answer API, you simply give it a PDF file and say,
compress it for me in ten sentences and you will get ten sentences output.
Where does it run? We don't know. It runs on an Nvidia server somewhere.
See here. Launch locally or scale with Kubernetes. So that is how the containers come in.
So you already have containers. You have the microservices.
And it's all running on a GPU cloud. Well the last thing I want to tell you is to go here and get back and say Nvidia name Blackwell.
So Blackwell is a new processor.
And for Jensen, Huang's dream is, um, people random third party people will buy will buy Blackwell GPUs and build these clouds anywhere on the web.
So you can ask machine learning developers will write these name micro service scripts and run them on somebody's GPU cloud.
That's really the whole idea. So you don't have to own costly machines or anything, right?
Wonderful. It's all brand new. See this? What a month old.
Trillion parameter scale models. So Blackwell is, uh, shockingly amazing what Blackwell can do in one second.
A standard human being, a numerical calculation. A standard person like you and me with pencil and paper will take us 57,000 years.
Really? You can go to that if you want. 57,000 years from one human being calculating every second of your life with what it can do in one second.
It is shockingly fast, obviously, and they are just getting started.
In Blackwell there are two um boards and each board has all this course GP course.
Right. And there is a high ten terabyte per second interconnect.
So it's 3D, but why stop at two. You can have many more in the future.
So chips don't get built sideways in 2D because then you have the heat limit.
But now you can actually make them build in 3D. Okay. So it will be like many, many, many times faster.

Right now our GPUs have probably like a million core or something, but they can have trillion called GPUs.
Holy [INAUDIBLE] [INAUDIBLE], right? 60 trillion core GPU.
I don't even know what you do with them. Select trillion core GPU, the things that are being discussed.
See that this trillion transistor. But, you know, someday.
Trillion. Of course. But that's already you know, we talk about it.
And this is crazy, right? All right. So, um, back to what we're saying.
So all of what I tell you is like going in all these good directions because any of
freed of programing languages have to know just simply what the API does and just.
Right, right, right. You know, a language, translators just write anything.
No need to go back to the basics and say, you know, I have to write transfer my code by hand.
You can understand it just for fun. Several slides, you know, not too much.
We'll go through this. Okay. Some more things in all these neat things I'm going to tell you.
But they are somehow related to our wonderful. And in the last few years it's changed so much.
This course has been taught since 2014, by the way, and mostly not much changed,
although Google obviously made improvements to search engine every few years.
Okay, but after November 2022 and Jupiter dropped suddenly, it's accelerating in so many directions.
So next week I'll tell you like much more, okay? Okay, so this notion of a chat, an external memory rag is approximately what external memories.
Okay, okay. So the idea would be that the domain, the actual alarm,
the call alarm that you would call the call this one, this right here is not an expert on anything.
And the reason for that is OpenAI basically grabbed every script,
every piece of text they could ever find from anywhere even questionable for kind of right wing sites.
Okay? And there's no time for anybody to go back and clean it up because of massive volumes of data.
So it basically ingest, like, all kinds of crap, okay, even dangerous, horrible, violent, um, incorrect things.
Science is a lie. Holocaust is a lie. Earth is flat. It doesn't know, okay.
Just takes it all. So it's very dangerous for us to basically use it and trust what it says.
Right? Okay. That's on the one hand.
But on the other hand, if you make a vector database or even a knowledge graph or even a relational database with tables in it,
you can run SQL against it and suddenly the whole thing changes, which is I don't have to ask alarm times or anything.
I use it like a front end for natural language interface, but I tell it, do not answer the question yourself.
Go in the external database and if the external database does not have the answer, come back and tell me I don't know.
It's all in the prompt and then you start talking to it. Okay, so that way I will never lie to you because it can online.
That's why I call. Right. So that is what all of this is. This one is pretty cool where ChatGPT can actually be used with a graph database.
So again what can this external memory be?
This external memory can be a vector database where all your external knowledge is number in vector and stuff like that.
Or they can be, uh, these days so many other things that can be a knowledge graph.
I mentioned knowledge graph in this course so many times.
Okay, the Google knowledge graph tensor things because that is high quality factual knowledge.
The type Bill gates he will tell you when he was born how old because it's all in the knowledge graph okay.
So that is a classic thing we can use. But these days our choices are basically unlimited.
You can use PDF files and chunk that and vectorize it.
You can use CSV files and actually get tabular data.
You can also use plain text files.
In fact, one of the homework question number two is you will make a nice Json file with all of plain natural language words.
Okay, that can be used to do a regression. There can be some simple piece of text.
You can also use relational databases which are actual tables, and you can run SQL on them.
Because many LMS, especially for has an API, you can make external calls.
You can even make it run matplotlib and actually plot something.
So therefore there's no end really how cool the external stuff can be.

It's almost like you turn the tables okay for a while to produce some magic, you know, master everything or tell me about God.
But now suddenly I don't even talk to me. Now you're just manifest so somebody else has a better answer.
Come back and bring it to me is basically the idea. So this one is one of the best ways because it is a knowledge graph.
Want to show it to you. Here, right? Learn your graph schema.
Help you construct cipher queries. So how does it even work?
Right? The knowledge graph in this case is a program called neo for J.
It's a commercial product. That program comes with the programing language to go to the graph.
The language is called cipher. It is not SQL by the way. It's actually a lot easier than SQL when you see the SQL example.
So for example cipher is pretty easy to pick up. So here your query will get turned into a cipher query.
The cipher query would run on the graph.
The graph a lot of node information that's given back to L, M and L m can summarize for that you in plain English,
like what are the first five movies that Steven Spielberg shot, for example?
And if the Knowledge Graph has the IMDb movie database, it will actually answer your question.
That's pretty neat, right? In plain text. Okay, just ask it something. Okay, so that is magical.
All right. So Neo Forge, you can run ChatGPT set up in a capability defined graph schema.
Okay. We won't go through all of them but okay. So schema is simply table definition.
All right. So yeah they using IMDb. Let's see what question they'll ask okay.
Okay. That is that is what cipher looks like. The cipher is very interesting.
Nothing like SQL right. But like SQL you can do table unions.
There's a little union command. All right. So dissolves. Okay. So generating cipher queries which has a pretty that is the cool part.
You type something in English like a comment. And your comment will turn into a cipher query.
And the neo forger does not know where the cipher code came from.
A human could have written it. In this case, the compiler will like this.
You will generate cipher queries based on prompts. So look, do not post in explanations.
All of this is your telling. The LM okay. Okay. And then we can start.
Yeah. How many movies did Keanu Reeves act in? Then your query will get turned into an actual cipher query.
This one is cipher. If you know cipher, you can write it yourself.
But most people in the world don't know cipher. Nobody can or isn't so cool.
And it will actually come back with the answer. That is the crazy part, you know?
So fantasy. And then it'll come like here match play match.
Image by the author. Optional match. At some point it will turn into actual English.
Okay. What are the most popular common? In this case though, it looks like they're not showing you what it does, which is sad.
They're just showing you how, um, your plaintext can become a cipher queen.
But the real action happens when the cipher query is run by neo, for example.
You get the results back, which at the end of this article we can see already.
You can see it's so powerful, right? Uh, look at this one. Okay.
So in here you asked it something that it cannot answer. So it came back and told you I basically do not know.
That is great otherwise. And without all this extra stuff will tell you some random crap.
Okay. Frequently wrong, but never in doubt.
That's characterized okay. Or even occasionally wrong, but never in doubt.
If you ask me something, I don't know. I'll be honest and tell you I don't know.
Because otherwise you will ask me something more complex. I can only lift a certain point, right?
Easy for me to say. I don't know, but does not know how to say I don't know.
All right, so this is one example of many, many, many. I'll actually summarize it for you.
I'll just say ChatGPT. Okay. With, uh, near forger or even with a knowledge graph.
Then you will see all the rag applications pop up. This is one of the best ways to do rag.
So now this is documentation from Neil for Jay.
Okay. Might as well say. For over a decade, Neil for Jay has been helping the world make sense of data.
Today. For over a decade, Neil for Jay has been helping the world make sense of data.

Today, Neil for Jay, the leader in graph database and analytics technology, is unlocking new possibilities in generative AI.
So from that point of view, more people can use plain text squaring as a public pre-trained LRM with your own data subject to your privacy controls.
Reduce. It also means, to a small extent, people that have jobs as cipher programmers.
There might be fewer of them in the future because we don't need cipher programmers.
There's always two sides. Okay.
Powerful combination of deterministic facts and probabilistic conclusions and enhance explainability and transparency through.
That's the great thing about Knowledge Machine for getting from say,
why does this have to come from explainable tell you the node that give you the new forged, scalable and flexible technologies?
There is a big problem with standard machine learning because it does crazy things with the neural network.
You have no idea why it made certain decisions, because if you look at neural network, all you see weights, which are numerical floating point values,
okay, they don't look like anything humans can understand,
but a knowledge graph will point you to that node and say, look, that node called Bill gates.
It said born in Washington. That's why I told you that. So you can trace errors back to frameworks like Lang Chain, vertex AI,
OpenAI and beyond democratize the index the world's information while setting a new standard for AI accuracy, transparency and explainability.
Quote unlock new possibilities in generative AI today at Neo Forge.
So this one is not biased at all in any way. Neo actually has been around for like a long, long time.
So you can do tutorials. Okay, again, say you want like a tutorial, just simply say tutorial.
Everything is simply a search a way. Cool.
Even this one. Starburst data. Okay. Like here, create another one.
Um, yeah. Okay. So this one chat, you know, again generate cipher queries.
Previously that article said, you know, what is the deal. But here they actually show you this.
You can see it's Java right? It's a Java. No it's Python.
What is this Python. Okay. So um very interesting right?
You can write all this code and actually run it probably on GitHub somewhere. But in the end it's searching through a graph like that.
Okay. Beautiful. Okay. So we can move on.
Uh, c r right there. So we don't need to know cipher anymore.
Likewise we don't need to know SQL anymore. You know, here I told you, you can go and have, like, a SQL database.
See here that one. So then who's going to write the SQL to turn your query into SQL query.
Right. The machine can do it. So you know GitHub Copilot two can all generate SQL queries.
So to a small extent people don't need to write SQL queries like in fact right here this one quacking duck.
It turns queries extended words into into SQL.
Cool duck DB nice new database playground.
Okay, there's a table, right? Employees department salary in order.
Actually, three tables and all those columns. So then some kind of, uh, uh, query you're going to make on that for example, like, yeah.
What is the average price of say there's a price column. And the SQL command is pretty simple, a big parentheses, you know, the column name okay.
But some random person on the street would not want to talk to me.
They don't know how to type of parentheses. Right. So then you can here just type the words like here uh ask question SQL prompt.
Answer prompt. So it is coming from a database.
You know, the the crazy thing is uh, like that the prompt is actually going to a database and coming back with the query.
Okay. So prompting here, uh, schema of my database in our database contains your helpful assistant.
Okay. So far, so good. Output a single query. Disaster.
Okay. Oh, right. To actually make sure to use tables, answer the following question.

Okay. Still not there yet. Um, yeah.

This one I got, though I don't know what all this is. It's supposed to basically, uh, you know, accept natural language queries.

Okay. Like, tell me the top three performing employees from last year to like that.

Okay. Or name me the most expensive employees. So, uh, that paper text to SQL capabilities of this whole paper for that.

By the way, there are so many models, they're all LMS. Each one is a different one, and they all do slightly different things.

And here they evaluate all of them against one standard data set.

Then you can compare the result like yeah that's gold. And then against score compare all the others meaning each element give you a different SQL.

Some might be right, some might be wrong. So it is not automatic that they're all working as we speak.

So that paper just compares them. Okay.

Anyway this fascinating that we can even go there because for the longest time in the world, only humans could write code.

But now suddenly that's not true anymore. That is what? Dangerous. This one is very interesting.

Okay, so one more thing that I believe a lot of people talk about this.

I am not personally sure, but, you know,

just go with what the model says is that the next frontier for large language models is not even, uh, just simply wrong.

I mean, right gives you a factual answers, you know, which is better than hallucinating answers.

But what is better than factual answers? Which is like, if I ask for $100,000 loan from BFA, what is the interest rate?

That is a fact that will tell you properly. It can do it. Okay, but what about reasoning?

What about things like, you know, it asks you why do you want the loan?

And then you, you know, try to justify it. You have a conversation with okay, so reasoning is beyond what any of these can ever do.

But that is the next frontier according to a lot of people want to combine LMS and reasoning so it can be more of an I am more of a human like agent.

You know, when you walk in the bank, you know, a real person can help you. You can talk to them.

So someday, when will you have an AI that is almost as good as a real banking agent, a banker, then that is what this is rethinking with retrieval.

Okay. Reasoning based because let's look at it again.

Your mileage may vary in all this way. Cool on the one hand, but then it definitely shows problems.

But you got to fix it okay Tencent in several. Cool.

Yeah. Rethink with retrieval. Knowledge retrieval like all of this.

Right. Faithful inference okay. So far so good. Yeah. See this notion of, you know, chain of thought prompting.

So this is where you have a conversation with the you basically input something and it gives you a value right.

Return. You would look at the return and then ask you to do one more thing.

So it's almost like you're guiding it. What to do. You know it's called chain of quote.

There's also Tree of Thought. I think I drew that one time. Okay.

Thought just means one thing, one little piece that that can do a chain of sort is you're making the alarm,

do a bunch of things only like a linear sequence.

A tree of thought might be, you can actually make that initial branch into multiple problems, even like a tree.

And then and then they can all put the results back together. Then we call it Tree of Thought.

It was a chain of thought. So, so many ways of doing all this chain of combinations of what is problem can do.

All right. So common sense reasoning it this way. You know like here that Aristotle uses a laptop okay.

So you know, suppose you asked a machine that Aristotle use a laptop. It has to do some little inference.

Right. It needs to know. On the one hand, the first laptop in the world didn't appear till like 1975 or something.

Aristotle was alive like 2000 years ago. He possibly could not have used a laptop.

Right? That's what you wanted to say. So a simple rag one.

Do it. You know, we have to actually put two and two together, right? We call it reasoning as obvious and dumb as that.

Silly little questions, right? If we did, would get it wrong. It might tell you some random thing.

Uh, the answer is 100% clear. There's no way in [INAUDIBLE] that should be answer.

Right. So the question is, how will you make the alarm set?

And the answer is you have to combine with an external reasoning module. Anyway, you can read the paper okay.
But that's a a fruitful combination according to a lot of people.
Okay. Uh, this one I have so much to say, actually. Hmm.
Okay. You know, I didn't tell you about Transformers. I know it's a little bit cart before the horse, but why not?
Um, when you do a transformer, as of today. ChatGPT, uh, Gemini.
You know, um, all these transformers.
Uh, they they process your query, and so they have this notion called an attention computation and much more on that next week,
a promised attention computation. So ultimately, a bunch of numbers that you would compute call attention.
Your query has a bunch of words. Your query can be possibly answered by what LM knows with a whole bunch of more words.
Okay. But some other words in what it can answer match your query better.
In other words, what you know, what words to pick and then how to write them.
And then how do I arrange my first word? Second or third word? All that is part of this computation call attention computation.
So attention computation turns out to be a quadratic problem, in the sense that attention means you're looking at a certain number of tokens okay.
And then you're computing attention with the same number of tokens each word against other words so to speak.
So there are n tokens and tokens as well. So so it's o n squared is how long it's going to take to compute one tokens.
Attention to some of the tokens attention. Yeah. Similar token writer okay.
So attention means the number between any two tokens okay.
So then that token right here you can find the attention with this with this with itself, with some of the words and those words put together loosely.
That's all it is. But I'm going to tell you about three things. Three vectors call key vector, value vector query vector.
But just wait and all that. But because it is an n squared problem, we cannot have a large context.
That's always been a problem. So context window is what is called even what GPT is for.
The context window is only like 60 4k or something. And I forget how big it is.
We can actually look at it okay. So GPT for context size.
But there's all kinds of ways to break through that.
Yeah. So I 2000 tokens. So we cannot have more than 32,000 tokens.
Looks like a word. Okay. So after you reach the limit of thousand tokens, it's got to dump what it knew so far,
so to speak, with your conversation and start afresh with a new set of tokens.
It's almost like had amnesia and forgot what he was talking to you about.
Think of it like that. So a conversation won't have this long. Continuity.
Okay. Because the token limit. Attention limit. So one of the early proposals from last year was this amazing idea.
Now I can tell you many more newer versions of these. Uh.
Called retrieval transformer.
So this one is a new idea, like a modification of the standard transformer architecture that gets around this whole context limit.
So I'm going to let you watch a little bit of this when I come back next week.
You guys. I'll tell you about transformer using 3 or 4 people's writings.
One of them is a person called Uli. When she only one. She's pretty amazing.
You looked up on LinkedIn. The other person is actually released y I'm sorry.
Uli one. And then actually the other one is AJ Ulmer.
Incredible. Great explainers and I have 2 or 3 other modes I'll use, but let's look at this really fast to understand.
Language models are able to generate text better than any software system we've had before.
And the larger the model, the more the training data it was trained on, the better the text that it generates.
There's a challenge here, however, in that these large models need supercomputers being able to run them and deploy them.
Vast amount of resources go into that. A lot of knowhow is required to get that working.
So the question is, how can we make these models smaller?
One of the leading ideas to empower smaller models have great text generation capabilities is retrieval.
This is the idea that the model does not need to store the world's information.

This one is a video called the Illustrated Retrieval Transformer.

They made a previous video called the Illustrated Transformer.

Just watch that video. It's one of the best explanations of Transformers work.

Into its transformer, by the way, is a black box that transforms some incoming tokens, like fireworks, into a new set of words.

Transforms tokens into tokens. Okay, loosely that is actually what it's doing, believe it or not.

See, a query is one set of tokens. Dances returns a bunch of tokens.

So a token. The token transformer transforms input tokens to output tokens.

That is why it's called a transformer. Amateur's. Uh. But I want to move on though, and show you like all of them.

Like the ability to, uh, to retrieve okay is very exciting.

Um, and it's. We saw a couple of these papers, so. It's so cool.

You can follow him on this one, right? This being one of them, the retro transformer, one of them web GPT being another one.

Yeah. So DeepMind is the one that came up with the notion of duel and has this very cool idea that.

But wait until I show you what they came up with last week.

A model with 175 billion parameters is crazy, and you don't need to see that all of the world's information,

it's, you know, imagine that that is the actual full blown parameters, that parameter.

What we mean is simply there are many neurons in neural networks, right?

Each neuron has a number of inputs and obviously only one output always.

But then you add up all the neurons number of inputs ten plus 25 plus 37 plus 48.

Keep adding for all the millions of neurons to have every input.

Add all the input count. That is called the parameter count. And so the bigger the whole neural network is, the more that parameter count.

So that is what this is.

So 185 billion is what was in Gpt3 and GPT for the parameter count the number of neural network input connections 1 trillion which is 1000 billion.

Crazy right? 1 trillion. Some Chinese large language models have 2 trillion parameters.

2 trillion. Even bigger than 84. Okay, like our 3.0.

I mean, it's just crazy. But in comparison, this notion of retro will actually squish it down to a small only 4% of 185 billion.

And so, you know, that is what the retro, uh, model is about. I'll go a little faster in terms of your model.

And but here's another example of a of a problem words and ideas.

There are techniques that we can use visual that actually not, you know,

need like all the parameters in general, the context in order to not be the most similar set.

Yes. So this is the retrieval part. It uses the prompt, the input like you don't so much released in right.

And then it goes and finds nearest neighbor. You know I mean this I cannot I will tell you a million times it's all about nearest neighbors okay.

That's all it is. It's a sentence nearest neighbors. And it will find the right answer for you.

Uh, 1984 and then 2021, we there are two Dune movies that you and possible prompt as a, as a prompt.

But it also gets this relevance okay. So that is one architecture called retro for, you know, break around the context limit.

Right. But let me tell you a more, uh, infinite attention okay.

So attention has a limit of sorry. Okay. Tokens I showed you are, you know, 64,000, maybe double it.

What about no limit at all to how much attention you can calculate?

Last week, Google made a new thing called infinite attention.

Last week. Infinite attention.

Yep. Internet text. Not even last week.

Yesterday. The [INAUDIBLE]? Okay, so, uh, it is crazy, right?

Context. Windows. Context. So cloud has 200,000 tokens.

So they all have, you know, some kind of thousands, right? Not even millions. So what if you break the limit forever?

So like. Yeah, right. The more context means you can input more data.

And again remember the previous conversation is basically limitless okay.

Like a human your friend can remember what you talked about last week. So you need tons and tons of memory.

But now suddenly you're able to do it. Because you actually compress past context.

If you say, how does it work? Okay. The older the context is, the more Google compresses it but never deletes it.

And the newest context can still pay attention to all of this older context.
So attention is never thrown away okay. So it's increasingly compressed. So a clever idea to extend is smart.
Let's see what they say okay. Okay. One day I got some [INAUDIBLE].
I don't keep up with the bleeding edge. I find all this fascinating. So one of my missions in this class is pass it on to teams.
So never be surprised by anything. You shouldn't hide seeds.
Okay, since another infinite context transformer has come out and it is called leave No Context Behind.
So context is simply, you know, the retention window is called context. Usually you delete the context, add a new one, start a new one.
But now you can have a new one. But you can go back to all the previous contexts increasingly compressed.
That is basically the idea. So they call it infinite attention. It's very cool.
It's called infinite attention. Very, uh, very strange name.
Um, yeah. This is the paper title. That's pretty cool.
You know, it's not like we have not context transformers at Delta.
That's exactly like this. Yeah, that's a meaning. Um, so I think it's just to stabilize this.
Okay. So papers, like, uh, videos like this are very cool because that paper might not be easy to read.
It's basically written by ML practitioners. Assume a b, c, d okay. But then videos like this break it down.
You know, draw nice diagrams and explain so it makes more sense to people like us.
Okay, there is the q v that I talked about here okay. So in the end it's all about keys and values and then queries, you know,
and you compute dot products between them basically call and then rank them in the softmax operation.
So just before that so it tells you how transformers work. You basically just chunk up your sequence into these segments here.
Yeah. Now what we're going to do is we're going to have a hidden state of some script this max up.
But if we have a just see this a key query again okay.
So much linear algebra so much well compact notation but it's not super hard really checking all the keys.
It looks like a long time to process internally because summing up and then using that to anyway say taking the whole separate part, we can skip it.
It's called off. So that is one more way to break the whole context problem.
There's a different way. The n squared attention.
One more way is we talked to this notion of transformer is fundamental to all of this that transformers are the feature.
This is a transformer architecture encoding. Right.
But suddenly but a couple of weeks ago, there's a new architecture that goes back to an older version of all of this, called a recurrent neural network sin rnn.
You know, these networks that basically feed back like a little loop, you know, like a for loop.
We started we moved away from all of that when attention paper came along with Transformers.
People said, wow, audience obsolete. But now there's a new architecture that goes back and uses RNNs.
It is called member sum. Member does not have quadratic attention computation.
It has linear attention computation. Suddenly you go from n squared to n.
So that is also a great way to break through the barrier. Some number uh number transformer.
Although, you know, you don't need to call it transformer anymore. But let me show you linear sequence modeling.
So linear time right there okay. And that's December 2010.
It's very cool. It goes back to R and it's like I told you Albert girl Carnegie Mellon very nice Princeton.
So this is an academic paper for a change. Yeah. They call it a selective state space model.
You know, as some, I suppose, a transformer model.
Great. So please read and come up to speed on this.
It's like three months old mumbo. Explain. So again, you know, the idea is the whole point is you can compute like more, more attention.
Really? Okay, so, like, your attention isn't all you need.
Great. Okay. That's all. Attention, by the way. So everybody's paying attention to all the other words, including itself.
Okay. Okay. So and then that is how Transformers work.
But then in Mumba they replaced actual transformer blocks with this notion of state space models, which is actually linear.

Okay. So we can ignore the details for now. But there it is. So it's basically going to allow you to transform like longer chains of inner input.
Quote. And then as if Mumba is not enough you can now Jamba like Jamba Juice.
Why pick between you know Transformer and Mumba. Why not combine them?
Hey, welcome to Jamba. This thing never stops, I can tell you.
Okay, well, it hasn't guaranteed what any of this is. Next time you look, it will be something else.
A hybrid transformer language model. Look at how many authors.
Okay? No longer. It's just one person. Like a whole team.
Maybe the whole, you know, a 21 labs. Oh my God.
Okay. And then they usually publish like a nice little paper like this. Right on.
And then they give you the model. Okay. You can actually download like all this and play with it is so neat.
Blog post white paper 2021. Yep. Okay.
And then you can try all this tonight. Pip install say mamba.
Send them casual. There you go. And then. Yeah, uh, you should have some kind of GPU and then do it.
And then you can actually try like all of these this like some data sets. Cool.
Okay. I think I want to come come back here though. So this is all basically about memory and or external augmentation.
Okay. Um, how to have longer and longer chats, basically infinite chats.
Okay. Uh, okay.
So this is where the whole Asian stuff comes in. Okay.
Today we call them Asians. Okay? But even a year ago, it used to be called origin pity.
And the name still exists, by the way. And also, one more call by which umbrella pronounced baby.
How baby? So, baby. Uh, orange.
Repeat. No matter what you call it. It's all an idea that the L does not have to solve some very complex problem with one big problem.
But instead you can chain like, little piece of, you know, problems together and then make a chain of them.
And so that is an autonomous. Then you can make the l m go on to each task independently and come back with a full answer.
So your whole answer is the the full big problem. If you had one big problem.
Otherwise you break. You probably do subgoals. In the real world, when you solve any complex project.
The first thing you will do is break it down, right? It's the same idea.
You call them subgoals, and then each subgoal is achieved by your different agents.
They can all come back and pull the subgoals together and you will finally get one goal.
This, by the way, has been tried for decades. Okay?
It never work because the agents could never combine whatever they could find, so to speak, because that required some high level meta reasoning.
Each agent can go off and, you know, do what you wanted to do, but then they cannot autonomously combine them.
You have to combine them. But then here's the idea of GPD.
You can actually try to make the AI itself combine all the results.
So it's hands off, you know? No way. All right.
So this is like again about a year ago I checked.
But in all of this right up my videos okay. Yeah. What is GPT select like here.
Right. This one. What is it?
Okay. So, uh. Yeah.
Designed, um, an example in this. Yeah. See like here increase network.
Go Twitter account, develop and manage multiple, you know, random assets classes for business simultaneously.
So you could ask it for like a business plan in this case a fictional business plan.
But the idea is it needs to know so much about how a business runs and come
back and give you a recommendation about how you can grow your business 20%.
You ask it very specific, like a business analyst, you know, what do I do?
It's supposed to give you recommendations. That is what it is supposed to be.
Again, your mileage may vary somewhat or someone may not want to move on.
Okay. It's fascinating just that you can try it now. Okay.

So all of this should be easy for agents. So that's why the word agent actually started to, you know, come into our LMS.
Although the word agent is very old Java has Asian programing built into the programing language.
So agents are not new at all. But we started saying the variation with the jeopardy context.
Say like yeah right. Thoughts, reasoning, criticism. In other words, it'll tell you.
So when you say do something. It's almost like it's thinking. It's typing.
Like what you see at the bottom of what I highlighted. That way you can actually catch it and say, no, don't go this way.
So you can basically give it feedback, right? It's telling you I'll search for upcoming events and you say, you know, don't search for your answers.
Look in my recipe database. You can guide it okay. That's the whole idea.
Okay. So then it's the notion of reasoning or call this. It can also criticize its own plan and stuff okay.
Yeah. Like yeah. You can read this. Okay. Why not just it.
Eight seconds. Yeah. See here.
It actually tells you how you can increase your business by telling you like what to do.
Browse websites, save important findings, gradations, continuous review.
You know, apparently coming up with a plan, so to speak.
And then like in coding, you can also do very interesting things like, you know, help me clean up my code.
You'll reason about code size you the word thinking think read the basic now
reading the existing code in basic math.py to write a better version of it.
So that is the agent's output action result.
It's very structured, obviously very scripted, very orchestrated, very good running the improved but still something to obtain the output.
So then you can use it to improve your code and give it your code and say, now tell me you know how I can better my code.
All right. Um, so how to use it, you know.
Yeah. And then. Likewise, baby, I might as well Google that baby.
Yeah. Uh, air power task management system.
And I see right there. Is there any Chinese translation of that?
Um, so how do you use a script like right here. Clone it.
Pip install. It's all Python. So. And finally you say Python baby API.
Please try it because these are the beginnings of agent programing okay.
And then you can run them in Docker as well. So that makes it very simple. So cool.
All right. Moving on. Um yeah okay.
So this one is also very interesting. Imagine having uh.
Almost like a role playing game where you have multiple agents in a run in a visual 3D interface,
and then imagine that they're asked to plan a birthday party for their friends.
They'll all get together and plan. A birthday party is all to I but to solve today.
But the planning is really very cool. It's like sims, you know.
That's exactly what it is. Okay. Generative agents, they look very cartoonish.
Obviously, nobody thinks it's all real, but what they do in this little virtual world is actually real.
They have conversations with each other. Uh, demonstrate populating a sandbox environment reminiscent of The Sims with 25 agents.
They all have to autonomously cooperate and achieve something. Well, not bad.
So then the next step from this is what 3D.
You can make them be in VR, Nvidia Omniverse, Photoreal GPU, and suddenly have 3D people that are equivalent of all of them.
So never to never basically dispense any of these. Like never laugh at any of these.
Take them very seriously. This is crazy, right? Look at this. These are Asians, okay?
John says to Eddie, good morning Eddie, did you sleep well? Good morning.
That yeah. Some great Asians are having autonomous conversations with these other emergent social
behavior scenes where the scientists find that when you have these agents run around,
they do some new code things like humans might do, like they might greet each other in a hallway or something.
Okay, that's not programed in the system. You can find new behaviors that emerge.
We call them emerging social behaviors. Brushing teeth.

Waking up is comical at some level, obviously, but like I said, you can make it be 3D, almost like trivially.
Uh, what are you looking forward to right now? Isabella is excited to be planning a Valentine's Day party at Hobbs Cafe on September 10th.
Oh, okay. So an invitation. See that?
All of this, huh? Interesting.
So from plain text generation, now they're having conversations with each other.
You know, again, like, they might be absurd, they might be cool, but it's really something new that the world has never had.
What what is common to what I told you before is planning, reasoning, reflecting on like what your reasoning about.
These are considered higher level human capabilities. Like your cat is not supposed to reason through the very complex, right?
But now suddenly limbs are able to reason. Okay.
That is why the word generator isn't all that comes in. Okay. This was a pretty cool paper.
And then. So Isabella is actually planning this party, and she's talking to all these people,
and then this person is saying to that person, Isabel has invited us to a party.
It's all automatically generated, all the conversation.
You know, there is something there for sure. Wait.
And then finally it all goes back to Sabella. I heard you're planning a party.
That's pretty cool. But relations. Okay. All right.
So how will this all become 3D? You know, because we have so much time, I can tell you.
Unity. We are agents.
So suddenly. Machine Learning in Unity interaction tutorial pass through.
How many agents can handle 10,000 agents is pretty cool.
Gauge interaction. Let's try that one. It's a little old.
No, I want to do a. Hmm. I'm going to say character agents.
Okay. Adding a character controller.
Okay. Hmm. I am welcome back to. Nope nope nope.
Okay, what about this one? Salutations, my friends.
That's exactly what I meant to show you, too. I hope the fountain.
So we're able to take our 2D that we get from the previous script that I showed you and actually make for real 3D.
Take a look. And your VR glasses. It'll actually feel like they're in front of you in unity.
You may have seen this piece of research at Stanford University working with some
advertising company where they have software agents simulating believable human behavior,
milling about and interacting with the world and each other, forming opinions and memories, and reflecting on those memories.
Deployed in Heroku and make dynamic a ChatGPT.
So now suddenly you are an entity and how to bring it in.
Unity has a chat interface which is crazy to manage it. So you can type the link.
You're a wise old man that only speaks in high jump back into unity.
And use the package. This is classic video game programing interface called unity.
Okay. This unity, this unreal from ChatGPT.
That's a crazy uh, programing language community has called unity script.
It is not C-sharp. It is not JavaScript. It's some strange hybrid between these two class for community unity string.
Okay, I'm going to keep going because you will actually see the problem.
Okay. Uh, okay.
Um, okay. So now your agents as you can see, are like very simple.
They're little capsules, but they're in 3D onces. So the next step is for agents to become photo realistic medication.
Me of all great names, are you take your face to become an agent in the 3D world and interact with your friends.
Um, it is. It is kind of limitless, you know, just keep going. Okay, so here we can.
So that took the 2D, um, world that you saw in the other paper and just go on adding code.
Uh, that doesn't you look at that. So that is weird. Can you suggest somewhere judicious that I can make that feature okay.
Talking to each other? Great. So our proof of concept, you said when I was going to I don't memorize all this dialog and

click on the link.
It was like I randomly write. It happens to match.
Okay. I mean, that's neat. See, all of that is automatically generated text between these two characters.
Once again, you know, it's pretty weird, really, because it is so easy to make this into a real character in this, into a shopping mall.
The text boxes, they are clearly to, uh, two wide here, but this is looking.
So then the code for that is somewhere available, I'm pretty sure. Okay, look right there.
Text unity wrapper. So you can then go and download this code and actually run it yourself.
So this is actually right there in the chat jeopardy for the unity wrapper.
So you can type chat jeopardy prompts in unity. And maybe one last thing.
Because if you ask me this looks nothing like real. How can we make it photorealistic?
I can show you. Nvidia Omniverse Nvidia omniverse.
Video omniverse. So then you will know. But the power of GPUs what is actually possible.
So that should be pretty easy. Check this out.
So then those little capitalizations can suddenly my name be that little Z character in 3D, in full photoreal rendering right there.
I'm here to talk to you, that's all. Like not photograph, by the way, CGI rendered pixel by pixel.
Let's first go over. It is stunning. Exactly is Omniverse. For those of you unfamiliar with it, Omniverse is an entire series.
Your characters can be floating in the little pool or driving the little cart.
Omniverse helps you connect different apps together.
Basically, you can collaborate with large teams and work with USD so unity can output data which will go into a domino.
Dominators. Omniverse is like a platform for integrating assets, which is basically 3D data.
It can be characters, it can be objects, it can be smoke. Let's take a look.
Look at that. So Jensen is the CEO of Nvidia.
So that took his kitchen and actually assembled it to an Omniverse.
That's so cool. We can use Omniverse to create and essentially move the mouse.
You have no idea that this is a real video, right?
It is truly stunning to allow. So our agents from the two papers we saw before can actually be running around in here.
Uh, final images, videos, cinematics, whatever you'd like.
There's also that could be an agent which can help you put together so that talk about birthday parties and all the, like relaxing trees.
Okay, Nvidia website for that. Uh, going over just exactly how you can use some of that software for machinima and a really cool example of putting together a short cinematic in about an hour or so.
Again, your AI characters can actually do this. And, you know, a face, which is an AI facial animation tool that helps you convert simple sound clips.
Wow. Awesome animations. So this is my world of 3D graphics, but you can see, right?
It's so for the real one that until you move the mouse and you know, like that, change it and you don't know that it's,
you know cuz it helps you to work reverse it means you will no doubt that the future world update can be 100% photoreal creating in 3D,
by the way, in VR. So this way you don't have to open your entire scene, you can just open up.
Okay, so this one is just all about Omniverse.
So tell yourself that any generative AI stuff, like all of these can go into the Omniverse and really be photoreal 3D.
So let me tell you more. Okay. So again, where we're going with all of this is so this baby here that I mentioned, okay, so I can skip it.
But the idea is people are actually talking about a whole AI operating system.
In fact, one time I wrote a OS and I go to the paper. Okay. So people think that in fact windows is making an AI laptop come out in the fall.
So the idea would be that you have the core operating system with all the OS calls, and you have the high level things like chat, translate,
all the things you want AI to do when some kind of a layer in between that translates all the calls directly to a GPU underneath.
So we call it the computer system actually. So that is actually what's happening.
You have LM and tasks and then some kind of database to hold all your context and past training data.

All the. Right. So that is where the notion of blank change started to come.
So long chain and um llama index. These are two names.
There's no long chain. By someone called Harry Chase chain.
And then this thing called llama index. What is common to both of them is that they're both Asian programing languages.
That is how you program all of these from chains, for instance. Okay. You can do either one.
They both work pretty similarly. Long chain came first, then afterwards, uh, let me index them afterwards.
For example pinecone hashes intro and then pseudo language one more people don't talk about this anymore.
This has been replaced by llama index I think. Okay, but long chain is fully alive because deep learning that I the company that uh,
Andrew owns, they make videos about long chain with the founder called Harry Chase.
So our friend is back today with what will be a real collaboration.
Who's gonna become an expert on what is called long chain.
Now, long chain is large language models and web search and all these.
Interesting, right? Who is Olivia Wilde's boyfriend? What is his current age?
This is weird. What is the age range to 0.23.
And you can do some things okay. Okay. By the way so it doesn't know the age right.
So this is what a chain does. It says, you know I need to find its age.
It's like thinking or allowed okay. And you can see what it's doing. Like crazy different things that we can.
So let's get started by having a look. Okay so pip install let's change the toilet for components, get lunch and get everything planning.
So we have prompt templates large language models.
All right. So in other words it's an Asian programing language the okay for this.
So you asked this question right. It's doing all this. The answer I put in like all of this.
And yes it actually found its age raised to 0.2. Use a webcam right.
If it goes to the websites component type system preferences need to change this.
So there's hope. So there are a few. That is a large language model called Slam.
There are so many okay that you can download obviously maybe it's getting some hugging face.
More hugging face has a repository of 300,000 models, 300,000 whatever you can,
you can tackle them anymore and you can name them and go and download that onto a machine from long chain or the model is actually that.
So you ask all these questions, right. And it's going to come back with an answer like all of them anyway.
So when it does a chain of them you're going to find one, two, three, four.
Someday you can have a connection between all of them. And then if you want to use those reasoning for you.
What did you have? Okay, so on number one I want to show you one more time in case you forgot.
If you go to deep learning that I. They have all these courses, right?
They have long chain courses. So please, please, please, please learn them.
This is so cool by the way. Just like input, you know, for the AI quantization.
That's the other thing I forgot to tell you is, uh, you can take these, uh, the neural networks, weights and the weights, usually a floating point numbers, but you can convert them to more bit crushed four bit weights or even two bit weights, which is pretty small.
So you lose accuracy. But they're also so small now they can run on a laptop or your phone.
It's just like a small weight model. Okay. So I want to show you quickly the part that about length chain index is also a programing language.
So lama index I told you Lang chain, there's so many, uh, knowledge graphs, open source models, prompt engineering.
Lama find the language chain one and that's the pinecone one marks.
And finally this one. Okay. So just watch, like a minute.
According to GitHub, recent reports on the state of the find the people that work on also lead guy for lunch.
So the inventors name is Harry Chase. But this number of applications move from exploration to application.
Many developers, including many web and mobile developers, once Billy Jones, have their apps in JavaScript.
Lang chain is an important orchestration framework for generative AI and is used by many teams for coordinating workflows using large language models.
Happily, it supports JavaScript. I'm delighted to introduce our new short course built um apps with Lane Change.
So when you actually go to simply the introduction, right when you go through the whole course,

they have a virtual machine running on the side with the actual code, like a Jupyter notebook environment.
So whatever that talk, you can actually click on even modify a few things and try it out.
So you are not just watching passive video, you playing with the software.
So please do it. Okay. That's it for this course okay.
Just give it away.
Is a founding engineer at Nine Chains and also lead maintainer for Lane Change as he's collaborated daily with JavaScript developers.
How to integrate LMS into main. After this, we'll get back to our slides and I'll show you the creation.
Basically, it's been a pleasure working with you. I'm looking forward to showing you.
Changes. No, that's actually separate.
You know, you can compile that website if you want, but initially it was Python and imported that the JavaScript, you know.
Yeah. But definitely can go into WebAssembly and incorporate large language models into your applications.
We have seen increases. So that is the other trend in which you want to know.
Microsoft as an example. They have all this algorithms running in things like PowerPoint, Excel, uh, word in video, things like Grammarly.
Right. That can fix text. But now you can ask word. Start me a paragraph to send to my professor.
It will actually pop the text right there. And word, you know, so they want it to be actually part of like your day to day life, you know?
Likewise, in a spreadsheet you can say summarize these two columns and make a new third column.
You don't need Excel spreadsheet programing. You know, Visual Basic. Nothing, right? You can just do plain text.
So they want to integrate all this functionality. That's what I mean by saying, you know, build application, trust the JavaScript to build with LMS.
And we're really at a unique moment in time where technology that was very transformative in a small
number of machine learning experts has now become accessible to the broader developer community.
I'm excited specifically for the new opportunities for web devs to leverage their unique skills and creativity to build great things,
like changes to seeing about $1 million a month and after less than a year is approaching $10,000 10,000 stories.
In this course, you practice using the fundamental building blocks of that change to build a very popular class of on application retrieval.
There you go again. Right, Alex. And you'd like to make that data more easily.
Queryable for modern research. The steps in an our workflow will typically include loading the documents, searching for the most relevant passages,
and then prompting the alarm to synthesize this data in an information set,
goes and finds external data and comes back and summarizes meant all this much more efficiently using language modules.
So you can do the programing yourself. You can chunk, you can index, you can search, you can but lunch and makes it trivial.
So just a few calls, you know pinecone has a library. Also pinecone has a library called Kanopy.
So if you use the pine cone vector database you can do the chunking, indexing, searching, you know, top five reranking.
But then the Pinecone library makes the sorry, the Canopy Library makes it easier.
So there are writing libraries to make those calls you an easier. So does Trump's models.
All these restores and pauses? Absolutely.
And our goal in designing this course was totally different on the basis of building land based applications.
Right? Call logging, error handling, everything.
React to this course. And if you are a Java developer, I know.
All those courses that have a great, by the way. Okay. All right.
So I want to show you. Right. Okay. So just like man Stack, you know, all kinds of mean stack, right?
There's all kinds of stacks and, like, you know, basically up front and back in programing,
you can think of a new stack called the Opal stack open AI, pinecone like chain.
That is very neat. So Opal stack, we don't use that phrase too much anymore, sadly, but it's a cool idea.
So someone call when when you know she came up with it, like right there.
What is an opal stack? Open a pinecone link chain. Yeah, because they're all doing slightly different things, right?

That's got the knowledge base, that's got the whole Asian stuff. And then I can summarize all that for you.
So why not combine them and start using them. And so then you call it that okay cool.
So again there's actual code you can run here. Whoa.
Pinecone gives you a free API key, by the way. So see this?
Beautiful. You can learn about the API stack. And then this was just simply there was a course that took place last time, which is pretty neat.
If you live in or near San Francisco, you can actually go to all of this in order to meetup.com.
Some of you guys don't know about meetup. I'm watching the times I want to finish it all, but at the same time actually having fun with you.
If you go to Meetup Los Angeles, I want to tell you so Los Angeles is a pretty big place for meetup.
Okay, not I came before the pandemic. This was highly active.
The pandemic killed it all, but now it's back.
So if you go to meetup dot Los angeles.com, what is going to happen is you type words like machine learning lang chain.
You know, I am just trained and then you're going to find something.
I'm just going to say data science. Let's do more and more okay. Almost every single day there is something going on in LA.
Online data science, intraday data science curriculum, data science analytics,
job market okay women and data science just go okay, just totally out of scope.
Most of free this near for J. So what if I type?
What if I type graph db because I want to show it to you all. Then I'll type lms okay.
So graph DB neo for J advanced flag techniques instantly.
But that San Francisco the main one right there okay so and then what about like what is like LM I'm searching standard meetup by the way.
I mean in meetup you can search for anything. You can search for dating seriously or cooking right.
There's so many. So it's obviously everything. Right.
But then if you type something so specific like self-driving cars or something, see that fundamentals,
you know, monitoring, mitigation, entertainment lamps, lipstick all over the place you know.
So like pretty neat right Los Angeles what else should I type.
And then we can move on. Well, almost anything you want to find in here.
Drag workflow, drag applications. You know, just learn. Here's a secret little hack if you go to these meetings okay?
So find out who's talking like somebody's talking. Get their bio, look them up and see what they do.
Right. And then maybe find out a little bit more about what the talk might be about and what you don't know, obviously, but to find out, ask a question at the end.
It's going to be a much smaller room than this. But there are ten people attending. Okay.
And then just talk to the person, ask a question. The very next thing out of your mouth is, do you have a job for me?
And I'm serious. Yeah, cut all the crap out.
Click on LinkedIn or 10,000 people. Screw that. Okay, seriously, don't talk to any recruiter, okay?
Just to [INAUDIBLE] with all that. Talk to the person doing the work. The worst they'll tell you is, you know, send me a LinkedIn resume.
We'll be in touch, you know, or I don't have hiring authority. I'll talk to my manager, let them tell, you know, ask.
Just ask them. It makes it so easy to approach them. Okay.
There's so many I told you. Okay. One last one. Self-Driving cars, which are almost dead.
Although this waymo's in L.A. now, but just kept that as a group for everything.
See that? Actually, no. Not that. So green cars? Um, yeah, I don't know.
Maybe. Maybe not everything. Maybe type SDC. Right. I want to find it.
I guess not. Okay, so finally, you know, I stopped working, which means time to move on.
I want to show you what else to show you. Yes. Named entity recognition.
So talked about never before. Never is how you build knowledge graphs okay for anyone knowledge graph things connect to other things.
Again like, you know, Elon Musk SpaceX and or Tesla. All right.
You need to know that when you read like some standard article about Elon Musk, while he's an important personality, I got to pick him and also pause the sentences and talk about SpaceX and realize SpaceX is a business.

So automatically create those graphs. There is a notion of named entity recognition.
So I can help with that, because the reason is I know so much about language, right.
So one kind of a transformer is called a bidirectional transformer.
That is what B stands for okay. So bidirectional encoding subverts a kind of transformer architecture.
Therefore we can use Bert for doing there.
And that means I can take any book about anything, any news article published and automatically extract those entities, because there is a lot of times done by hand in or by some pretty low powered, like, you know, simple language models.
But now suddenly see here, LMS have stepped in and said we can do a better job.
See this entities in a sentence or text, one word group of words, you know.
So suppose you say the name is bond James Bond. You should know instantly this an actor, you know a famous movie character.
There it is. And then person, location, organization, person, place or things that you talk about in 20 questions.
Right. That is what any loosely or. So what person in place thing can we extract.
So in this case Bert and Bert is simply a model that can be programed in any programing API.
In this case to use PyTorch. You could use Keras. You could use in anything else.
So there it is okay. That is pretty neat. That is able to take plain text okay.
And then at the end output actual nurse. Wow.
Through Bert. So in this case it just walks you through how to do it.
So all the text you want to read might be in some file and all the named entities you want to look for, which is not magic, right?
You have to give it like a list of things to look for. You know that you would know important.
So maybe that's in the CSV file. So then it will go and find stuff for you like that.
These are shortcuts for what kind of entities you want to look for in the NER.
You want to look for people. You wanna look for time, you know, artifact event, you know, natural phenomena.
Quote. And then will go and find it for yourself. Skip all of this for you.
And that is the raw data. And then that's going to turn all that into next.
So then you will actually get a knowledge graph okay is what is on it. Okay.
Yeah. For each data and giving you like all the Python code you can run when any training loop where the result is training going on.
Now it knows look for nurse there are we. Almost at the end.
Hmm. Okay, this is funny. You know, articles like these usually stop short of giving you, like, the final result, you know, just about.
Oh, well. Okay. But there might be others that you like. Go all the way and actually see what you did, right?
Okay, so then the next step up from named entity recognition is a whole notion of topic modeling.
Over and over we go back to classifications. You know, we think that, you know, something is classified.
How will you then know given a new document? Is it that article?
Is it chemistry. Is it biology? Physics. You know document classification, right.
That is one of the videos that I missed when I was sick. So we are the classification slides.
But now you can actually do you can do more recent studying by learning about Bert topic.
So Bert topic is basically an approach neural network topic modeling the class based term frequency inverse segment to consider.
So then what they're saying is they're using a transformer architecture for automatically classifying things like documents okay.
Let's call it Bert topic. So when you input a book it will tell you what topic or topics the book belongs to.
Okay. Let's call Bert topic. I'm showing you like, you know, a little bit of everything.
Right. And that is Bert.
Bert, by the way, is something that Google came up with almost and almost immediately after they wrote the transformer paper transformer was 2017,
Bert was 2018 or 2019, I think, almost immediately. Bert, by the way, is actually the very best thing that you can do at the transformer.
In other words, Bert is truly what changed things for Google. Transformer is nice.
But when you do the so-called bidirectional encoding by encoding simply means say you have a,
uh, and you say your sentence sentence a bunch of words, right?

Normally you would look backwards to all the previous words to know, like what context is coming up, right.
That is backwards, you know, attention computation. But there's no reason why you couldn't look forwards.
Also, in real life, when I am talking, you don't know what my next word is going to be.
Help me. Okay. Sorry.
You know, listen. Whatever the [INAUDIBLE] they want, right?
You have no idea what I'm going to say, but, uh, in the text, you already have all the text,
so I can compute forwards attention and also backwards attention to bidirectional encoding.
It's called Bert. Cool. So therefore, uh, and all that.
Okay, so might as well show you AI. NLP is how we get AI to understand us.
I'm not sure if this will talk about Bert into parts of speech. For example, barked has a root of bark.
So the there's just no sound of dog there. There might be other Bert videos I can watch you guys.
All right, so history of bird. Let me see. Uh, Google Transfer model was right.
And then, uh, Bert was 2018, so. Right. Okay. Just one year.
Okay. I have to Transformers, but was the coolest thing that came out.
Then they started using this in search engines. So like instantly that's actually what happened.
Take your search query and run it through Bert and like not answer.
And that's neat. This multi language query is obviously by the way the first transformer when you say it transforms input tokens in output tokens.
It was meant to do language translation. Input tokens were in English.
Output tokens might be in French. Because the world's hardest problem back then and still now is to do actual language, human language translation,
because two different human languages Russian, Tamil and Chinese,
maybe three of them right now, almost nothing in common in terms of how words are spoken.
But here's Tamil people here, Chinese and Russian, they're communicating.
Okay, so languages obviously great structure, but no structure across languages.
It means if I say something in English. Now come on in English.
How are you in French? Comma A-level. It doesn't you know, it's not a 1 to 1 word translation at all.
So that is why it's hard. Like how the [INAUDIBLE] are you going to turn A and B?
But, uh, Transformers could do it. That is a beautiful part.
And then they realized you can do it. OpenAI realized you can do a lot more.
You can generate text, you can do document summarization, you can look and, you know, all that came afterwards.
At first the goal was actually language translation, believe it or not. Amazing.
So OpenAI should be given credit for that okay. All right. So but in the meanwhile here it is.
And self-attention again. So again the animal didn't cross the street because it was too wide okay.
So obviously when a human hears the sentence the animal did not cross the street because he has to avoid the word.
It is obviously for the street. The street was too wide.
Not that the animal was too wide. Okay, it might be possible.
Some animal thinks, oh my God, that door. But you automatically know that the word it um, applies to street but not animal.
Amazing, right? That is what the notion of attention is.
So when you pay attention, when you compare attention from it to all these words, then attention would be highest for the word street.
Okay, that's the whole idea. Okay. So then you could do backwards attention, but you could also to focus attention okay.
So that's basically how it would work. All right. So what is Bert do.
You can do a question and solve this. Uh okay. This what's so neat.
So you can suddenly do Bert for actually NLP, NLP tasks, you know, am I doing it NLP course.
And Bert has been specialized into so many different Bert's. You know there'll be even more of this.
Okay. Anyway, so that is the whole big deal. Okay, so Bert near guide.
Topic modeling. So again, remember the central theme of this slide is topic modeling.
So how do you actually input some kind of unknown text. You know and then ask it like what topic it is.
Topic or topics you in plural I need to sign up. You can go read that Medium.com article.
I have a very simple $5 per month subscription and on my home laptop I will automatically work.
So if you guys, you know, go to like a, you know, incognito mode and try and use that read medium articles.

Just consider paying five bucks. Okay. Looking for that because then you don't have to do all the weird stuff okay.
Just click on it. It'll work. Yeah. I learned so much from medium articles.
Okay. There are so many implementations. I try many of them. As many as I can.
It's a good way to learn. So the five bucks is worth it. All right.
So knowledge graph construction again I visited this idea so many times.
Right. I went to a little while ago. But you can do a classic knowledge graph construction by using a lamp.
So now here's one more way. Plain unstructured text I input like a bunch of text.
Maybe a news article that I saw and turn that into a knowledge graph that is almost like bordering on magic, right?
If it works properly, that is what this is. Cellar lamps are definitely.
Wow, that is kind of weird. Please don't have that apostrophe there.
Okay. It is a spurious apostrophe. All right.
So then, uh, again, you got some data. Customer help center table somehow.
Support computers. Motherboard offer. You know what all this is? At some point, you're supposed to get in Knowledge Graph.
Okay. When I say work, it's because neo forget I haven't read this yet, but look at that.
It's a knowledge graph. Finally. So many nodes all get wired to each other, right?
So imagine getting that automatically. I mean, I haven't read this and I'm not going to summarize it, but it does look like, you know,
the whole article stressed to us somehow that GPT can help you get knowledge graphs from unstructured text, like a dream for a lot of people.
Okay, so recommendation engines, one more thing we can revisit at the time.
You can use recommendation engines everywhere, right?
I want to just, uh, you know, have some fun by telling you about this thing called TikTok's recommendation engine.
This is not AI, okay? It's simply recommendation engine. But the reason why it's amazing is, first of all, it's tick tock.
The recommendation engine is basically porn, and the whole world wants to emulate it.
Instagram wants to emulate it because it's so addictive, right?
Why is TikTok so addictive? They seem to magically know, like what you want to watch, right?
That is because of the recommendation engine doing its work. So the crazy cool thing about this is the reason why I'm showing it to you.
It is obviously knowledge rated information retrieval, but they use a pretty incredible data structure called a Coco hash.
Wow. A quick rehash in a standard hash data structure.
You have a bunch of inputs to be hashed to, like a table, like a hash table with some slots and slots.
Maybe the idea would be the hash function will hopefully take all the input data and hash it evenly, meaning all of the slots are equally occupied.
One of them is not too much occupied. That's called a hash collision, right?
You want a good distribution algorithm when you have one hash table, right?
Imagine now having two of them. So imagine having like, you know, for example, a hash function hash function here hash function called function.
It takes some kind of input and sends the input to here or new input here.
Next input here. Next input and output. So it is pretty good at you know evenly using all of your hash slots.
But what if there's a collision. What if the next incoming data is also sent to over here.
Normally in a hash table, what you are supposed to do is make like some kind of a linear list and say, sorry, there is a collision.
And now when somebody lands there, I need to search through all of them and find like what I'm looking for.
That's usually what you do, right? But imagine you are a cuckoo.
Cuckoo is a very interesting bird. Because a cuckoo bird, when the female wants to lay eggs, uh, the lad is too lazy.
Will not find. [INAUDIBLE] not make a nest for her. Instead, you know [INAUDIBLE] tell his pregnant wife.
Hey, come. Let's go visit the bird's next door and go next door and kick the bird out.
Push the bird off. Tell them to f off. Take over the nest and then lay the bird there.
Cuckoos do that. Okay, so then that is what this is.
So then when this new object wants to go there, new object will kick the other object off, like get out, it's not my house.
Then that one will then go to a new hash function, call f b and hopefully end up in a slot that is empty.
But if that also was occupied, that will then kick this out.

And you have to now go in f n going up somewhere else, okay. To play that game 3 or 4 times.
But it's crazy though. Usually it's resolved. So that means avoid collision by having one more okay.
And even more crazy by Tannen's did not come up with this data structure.
It's from the 1970s. Somebody in Netherlands in computer science wrote this paper and said, wow, that's a pretty cool idea.
The whole world forgot about it. But then suddenly, you know, ByteDance comes along and says, hey, we should use this.
So check it out. Okay, take it up. Recommendation engine.
You know all this, right? Some monolith I'm going to tell about monolith coalition less embedding table is what they call it.
Uh, high priority parameter synchronization. I mean, it's a very neat paper for, like, what it is, but I want to click on, uh, that paper.
So they put it up on archive. Collision less embedding.
I mean, it's not technically collision less.
It is minimizing collision because you basically eliminate the collision by throwing out what's already there.
I can't leave everybody's bite dense, every single one of them except this one photon.
Wow. Okay. So a bunch of times, researchers and write this paper.
Uh, where's the hash? There it is. See that Coco hashmap.
See that, say, comes in, and then it would be out, and then B would go and kick like g out.
And then I know whatever this is. And then that would go here and finally go here.
So they're all like doing this, you know, basically like dance musical chairs okay.
And throw you out. Very interesting.
This Coco data structure and then the use obviously so many other things as well for load balancing you know message forwarding.
And then they have this incredible, you know, real time obviously no prediction engine.
So they can predict like what do you want to watch. The whole thing is highly worth reading.
But I thought the data structure was pretty novel. So whereas the data structure it might be let me see if you can find it.
If not, we should move on. Uh, actually, you know, the best thing to do is to find Coco again.
Coco? Cocoa hash. And you are number 16.
So let's just call to reference number 16. That's not a hallucinated reference.
So number 16 actually exists. Um, our Pag and Fleming 2001.
Wow. Cuckoo hatching a long, long time ago. See that?
That's the paper with the Rococo hashing. 2001.
There it is. Denmark. Just simply, you know, this theoretical looking hash paper.
But then. Attention, people saw Golden that. There is so much value in computer science and math.
All that right? It was forgotten, you know. I mean, obviously us didn't do it.
Google didn't do it right or Facebook didn't do it. So ten cent researchers.
Wow. This is a good idea. We should try it. And now this becomes part of monoliths.
So that is the recommendation engine. It's great. So now you know tick tock.
And then Twitter's recommendation engine is also pretty cool. You know, again I'm not going I'm just pointing out what is available okay.
In Twitter you can search through tweets in real time obviously. Right.
And then people are tweeting so many millions of tweets per second and this is what they do.
So they start with a massive bunch of tweets that you search for and they filter it down.
It's like a two level search process, you know. They identify candidates that could be relevant to your search.
But it's not all of them are not candidates. All of them are.
And within those candidates to rank them and finally give them finally give you what you want to watch,
but also because of your profile, they also makes ads in and then other people for you to follow.
There's a blended recommendation okay. But the main idea is doing this filtering deal.
And then likewise Lyft also has a recommendation engine. Okay.
So then you wonder why the [INAUDIBLE] should lift a commendation engine. They're not recommending drivers for you right?
That's a match okay. Here. What they're recommending is based on how much money you spent in the past for things

like upgrades.
So they'll tell you right now if a small sedan comes and picks you up, it's enough 15 bucks to go to the airport.
But for five bucks more, you can have a big limousine. Do you want it? And how do they know what to recommend?
Your past preferences stop selling. Okay. That is what the recommendation comes from.
It's very clever. Obviously if it doesn't work, they wouldn't do it right.
There's some number of people that says I'll upgrade. Okay.
Here. See here. Like this one type module in all this visual highlighting like here.
All the prices might be in introduction. All the same. Um, what does the recommendation system do?
Um, challenge of our choice like this. Even for some standard like.
Right. I'll recommend all these for you if you want to share your right with other people.
Solidarity decisions you want you know of course it more desirable.
Yeah. And then some high end car. It even says quiet.
You want like a better cards. Like more money I guess. So you can always upgrade if you want.
That is the recommendation. Only 12 minutes and two more slides.
We can do it. Okay, so these three things have something in common.
I'll not go through all of the machine learning algorithms, but the reason why they are amazing is this.
TSA stands for T Stochastic neighbor.
Embedding. Stochastic neighbor embedding. Same as neighbor embedding is all about embedding.
Okay. And then there how can you map what's common to all of them?
On the one hand, the what are called dimension reduction algorithms like PCA.
You know, principal components analysis.
If you have a table full of ten columns of data, you only want nine columns of data and one eliminate one column.
The easiest thing is to throw it away, but you shouldn't throw it away because you are losing data.
You want to find a way to combine the data with all the other columns, right?
In theory, it works like this. Suppose you have two trees worth of data.
You only want to keep one D okay, but the data distribution when you it in 2D looks like this.
Looks like this. One way to turn this 2D into one D is throw away the y axis.
You think I don't want the y axis? So I will project all of them to one dimensional numbers.
Right? Then I'll take all of these numbers that I got projected, and I'll call that my data set that value, that value of x.
So value x you know. But then you are throwing away the valuable y.
Conversely you can project all of them to y. That means only keep the y value in x comma y throw x.
Okay. That seems like a pretty bad idea. Instead, here's a better idea.
You should look at the shape of the data and tell yourself wow.
Data is like an ellipsoid. I want to find the eigenvectors, the most important principal components.
That is one component. Clearly that is the second component. So I should use this measure axis to do the projection.
That means I am going to project like all the data from both sides.
Anything below that line would get projected to the left and then anything on the top right.
Then this new axis and all the points on them, right? That's the origin.
I read off all these points that are projected. That is my combination that went from 2D to one day.
That's a much better idea because I didn't throw anything away. You know, I found the best axis, right?
It's a great idea. Likewise, you can do this from 3D to 2D or 3D data, and it's fully clustered like this to the left side.
I can then take the smallest axis and project and turn them into 2D.
3D has become 2D, then 2D can become one day. I can start with 20 dimensions and gradually one at a time.
1918 1716 1514 1312 1110 98765432.
1 or 2. Usually we stop at two. So it's crazy that we can take 100 dimensional data and bring it down to two dimensions and see what it looks like.
So I told you, right? It's a 3 million dimensional image.
Like, what the [INAUDIBLE]? Right? How do you even know what the census look like in Transformers now?
50,000 dimensions. Because that's how many common words are in English language, right?
But we can project all of them down to 2D and actually see what the training data is like, what the transformer looks like.
It is amazing. So that is good reason to learn about all of them, because you might not know,
it might not do dimension reduction, although you might, but just for the explanatory power.
So look at that. That came from God knows how many dimensions, but you can clearly see the clusters, right?

Because it doesn't matter how many source dimensions you target, you went down to as few as two, even this one.
Okay, you cannot use PCA for all of these okay. Because data is highly nonlinear.
So but with things like DSA you can do nonlinear projection.
You can actually go along the shape of the data and actually project as you go along.
You know very neat things. All right. So again so this is how it works okay.
How do you project. Please learn. All these are not particularly difficult but they are so cool.
I'll show you TSA I'm going to Google it and you will see how neat it is. Okay.
You can even project like all of that. I mean, that's neat, right? You can untangle like, all of these and actually predict the monitory letter.
That's surprising that we can do that. Okay. Yeah. This was invented by Hinton, by the way, Geoffrey Hinton,
who was one of the three Turing Award winners for AI, you know, a long, long time ago, invented TSA.
Okay, I'm going to show you, uh, let's go here, t s and.
Just to show you how neat this is series, you just see cluster after cluster.
What is common to all of them is they all came from many more dimensions than two.
But yes, and you can boil all of them down to just simply two of them.
I'll just even say TSA Egypt. I'm just curious.
Okay, so these might be topics, you know, or words even that form clusters in multiple dimensions, but in theory we can see them clearly.
Okay. So we can ignore this. Likewise you map likewise DB scan okay.
That's what's common to all of them. In fact let me actually do you map okay then you'll see I'll just say you map you map dimension reduction.
You map dimension. You'll see something very similar.
See, this slide could have been a slider, but it's not your map.
It's a very different algorithm. And finally DBscan and something else called hierarchical DB scan is one more algorithm.
Look how complicated this data is in 3D. It's actually an airship right?
Things like PCA will completely fail. PCA means you're looking for one axis to project all of them.
There's no one axis. It's like a manifold. You go right back to manifold.
Euclidean distance doesn't work anymore. You need to project along the shape of the s along the actual surface.
And that is exactly what you can do. Uniform manifold approximation.
That's what. You know what? Elapse stands for approximation and projection.
Usually do it for dimension reduction, but you can use it for visualization purposes from any number of dimensions down to.
Okay, so DB is kind of similar okay. Skip it. Such.
Okay. Um, just random things about each other.
I told you about modules. You can skip that. You can search to Creative Commons license to say you want to.
You know, probably some kind of a manual. This is before stable diffusion where you can type some prompt and get an image.
You want to search for images that other people published. But people have copyrights okay.
And so they don't want you to use your image without the permission. But many people give their images away.
That is called a Creative Commons license. So if you can then search through all the images that have Creative Commons license permission,
then you can use their image in the audio without any repercussion.
Nobody's going to sue you. So that search engine amazingly finds anything.
Only that has a Creative Commons such, uh, license ready license.
That is what this is. So this is called open verse. Okay. So then you can go and look at open verse if you want.
But I just wanted to point that out here. So it's neat. In other words there's a search engine and you've never heard of right.
Yeah. I'll just say, uh, I don't know, oil painting or something.
So any number of people paint oil paintings, but many people put it up on Etsy.
They want to sell it to you, right? But what is common to all of these might be that they basically give it away and tell you what license it is.
You can go and read about it. Okay. That uh, here c cc you know, by NC all this means, you know, you have to just simply, um, cite the user's name.

Once you tell the world who made the painting, you can freely use it for whatever you want.
So licenses come in so many different varieties, and it tells you what kind of variety this is.
Credit creator, noncommercial decision. Or please don't make money with it.
It's neat right. So it's a custom search engine is my point.
Manifold search I told you it is a search that is not using just Euclidean distance.
Much more powerful than Euclidean distances. Again, there is a manifold.
So one last time, suppose you have a point here and a point here that could be considered close compared to a point here and a point here,
even though Euclidean distance wise they're pretty close. Okay. But this is one kind of data going to second kind of data.
Third kind of data physics chemistry math.
So physics point and a physics point is close. Even the Euclidean, they are not compared to a physics point and a math point.
You get data okay. So you cannot jump in air basically. Okay.
So you can do that. And that is a pretty neat paper that tells you so much about again embedding.
Please learn this because this is the uh, classic way to extend Euclidean search suddenly become non-Euclidean.
It's amazing. Okay. So and then again search through a graph.
So in a graph so many nodes. Right. So it doesn't matter physically look how far the nodes are.
It's just about what node is connected to whatnot. And you know that is one neighbor is close.
And in that is five hops is further on the internet.
And there is a graph search that is also clearly a search, but it's a nearest neighbor, but based on graph structure,
not based on any physical distance, unless you talk about countries obviously the near physical distance.
So in other words,
this node and this node are considered to be close because there's only two edges compared to that node and that node when you have many more edges.
Okay. Look at that. Yeah. So that's a graph search radius has search.
Also they use I. So I'll finish at 820.
So save the last slide. Always a constant this one. Any.
Here. There it is. Vector databases. So you can now create vector embeddings in Redis.
I told you Redis is classically a key value database. But now suddenly they have like I think a wrote here somewhere.
So they have a key value vector extension. It means now you can actually use Redis.
Why don't we do this? And it's complicated because I always save a little bit.
Our next speaker is Sam, part of the production with Hilda.
And that is our future search will be part one and are not.
Demos that show the value the sunny day make up our data set.
It's the same idea of cosine similarity. You seen this over and over in a slight similarity based piece of code on the very first slide.
Again, same thing. No matter what your data is, the all become made it through an embedding model and they all become vectors.
Then you can do similarity such a simple and that makes up the first half of this slide.
Three nearest neighbors are approximately. Hey um so ready it vectors show actual demos.
So things that that a separate point is going to return the nearest neighbors today and used request.
But you can see right in Python and in this case all as page image search.
You'll see that I can take a picture of somebody's shoe and then tell Amazon to order it for me.
Currently you cannot do that in Amazon by the way. Okay. But Amazon has an experimental search interface.
So there's Alibaba. You know they all need that okay. That is what the world can use.
So keywords obsolete no need for the title because you know I type water bottle 10,000 water bottles come up.
What the [INAUDIBLE] do I buy. You know. But instead I take a picture of a water bottle or so.
One minus the cosine similarity. Okay, cool. Archive papers um, Karpathy put out.
Once again, users are to be able to benefit from this vectors.
Okay, discord. Oh, this is pretty neat. Discord has many billions of images, right?
And how do you search? They use Tf-Idf. Wonderful.
Exactly like the Google search engine, right? Yeah. So then that's an easy.

And so you can read bulk indexing. They take all the words basically discord channels okay.
Let's start with that. And within each channel all the conversation they pull out the words and they use things like Elasticsearch.
It's very neat. So this one is a classic tf IDF, but discord happens to use it.
That's all. I might have time for the last one. Oh my god. Well, that's the that's actually the best.
I'm not going to do it again when we come back next time you can see what Google does with it.
You can go inside restaurants and walk around the tables and not just type the restaurant's name anymore.
It's all done by generative AI. So suddenly, you know a location based search will be like more in 3D is the point.
All right guys, so we made one last time. In the meanwhile, I will definitely give you a buy tomorrow.
My last thing that I'm going to do allow all for. Have fun with it okay.
All right. So one more time and if somebody wants to sing dance, please let me know.
You have one chance. All right.

Lecture - 6
I noticed that, DSI. Final and everything, right?
But this class is still pretty lean. Uh oh.
So I get to call attendance four times. You know, not just once, four times.
If your name doesn't show up, the first time will show up the second time or the third time.
All right, you guys. So, hey, what do you think this means?
Let's do a very symmetrical little. Little, uh, drawing here.
So, what do you think it means? Look.
Symmetry is a bilateral symmetry.
Symmetry. Symmetry. What does that mean?
It means one. Oh. What does it stand for?
Optimal memory transformer. Does it stand for. Something relevant to us.
What does it stand for? Really anybody?
Does not stand for tiny anything. Okay? Just one at a time.
Oh. What does it stand for?
Really? Such. No one's telling me what it stands for.
Okay. Yeah.
One more time. Exactly. One more time.
That timer. Is today.
So we'll do this one more time, including now.
And then we wrap everything up and you can all become such gurus.
Okay? And guru niece.
I'm not sure if that's a word, but no gurus. Okay. Uh, look at that.
Huh? Uh oh. See, right when I truly becomes powerful.
These things I told you last time, right? They're getting bigger and bigger.
And every time I looked at a brand new album that is massively large.
Cloud gives rise to cloud three, and GPT 4.5 gives rise to GPT five, the becoming multi-trillion-dollar parameter models.
I have all that to talk about today. Okay, maybe today the class is not for three hours and 20 minutes.
Maybe I say that every time, but every single time it goes right up to three hours and 20 minutes.
But maybe this can be a short lecture. You never know. So, uh, sit back and enjoy Waltz.
And this channel. Is Nichelle here? Oh, yeah. Right there. Cool.
So I'll definitely call on you one more time a little bit later, okay. Um, so we do have a couple of breaks.
All right, so that actually means that, uh, you can definitely come in, sing or dance.
Okay. Yeah. Cool. Made a little note.
So, yeah. This is your last chance to come and perform. Okay. In front of all your friends.
Why not? I'm going to go backwards.
I'm going to spend about a half an hour. So I have got all this planned out, right?
Uh, we'll spend about half an hour on some new things I want to show you,
and then go back to the very last slide from last time about location based services.
So, you know, search when you do location map search right now, it shows you a 2D map.
But that is highly about to change in a pretty amazing way. You know, the pieces already are there.
So I should talk about location based search being augmented.
And then uh, I have uh, topics part three, which is a segue from Topics Part two, meaning more things I want to tell you.
And then you'll be right up to Bleeding Edge 2024. It's really nothing that I have left out in all these slides.
I mean, literally, I mean, this came out 2 or 3 days ago. For instance, Microsoft announced this thing six hours ago.
Uh, tiny but mighty. So when I said bleeding edge, I'm not kidding.
So I'll definitely tell you about small language models as opposed to large language models.
I mean, l l m really? I started in November 2022.
Approximately. GPD is older than that.
GPD actually, they talked about it way back when, during the pandemic.
Jubilee 1.52, but nobody remembers all that.
Okay, so GPT 3.5 is the one that was used to make ChatGPT and the whole world suddenly, you know, like took notice.
It became a household sensation. But in in terms of information retrieval, that will be the biggest change that is

happening.
So we not going back to purely text based anything but basics will always be there.
Term frequency. Inverse document frequency is still a very great idea, but there is still keyword based,
useful in very simple context like code search maybe, but the bigger world will move on to things like generator.
I mean that is absolutely the way to go. So, you know, this field is changing quite rapidly right in front of our eyes.
So when I finish all that up today. All right. Last time I gave you guys I said Corgi was a very, very interesting search engine.
But gradually, you know, just basically fell by the wayside.
And some people think Google in its current form, which is heavily ad based, will also go that way.
But maybe what might come and save Google is the idea that traditional ad based search where to put ads on the top, and then the search, uh, results on the bottom will go away because of things like search summarization.
So suddenly search is not a bunch of linear URLs at all.
So they have to find other ways to basically sell to you. You know, in YouTube already, they have videos that you cannot stop with the ad blocker.
You have to watch the whole 30s, you know, so it's stuff like that.
But all that aside, I want to tell you, uh, search engines and our search engines and many will be definitely augmented by.
This is actually quite fascinating because he's one of the three, uh, Turing Award winners.
His name is Yoshua Bengio. And then just a little Japanese TV show interview about how AI, including In Search, including generative AI, can be,
uh, risky to people, you know, so that that word risk actually is it means so many things to so many different people.
Somebody who is very ignorant about any of this. The risk would mean for them.
Skynet, Terminator. Suddenly God's voice booms from the sky.
Sorry, humans are all slaves from tomorrow. I know infinitely more than you.
It's a very dumb science fiction idea that probably won't happen for a long, long time.
But other risks are things like a deep friction or pressure or such being polluted by, you know, bad results.
All of that real. So let's see what he says, okay. I want the whole thing.
Say, I'll play some. I'll just basically jump around, but it's actually highly worth.
I'll show you part two of this also, which is a more broad bunch of.
We're in Montreal, Canada. We're here to visit one of the global sentences for AI research and development.
Hello and welcome to deep. Yoshua has a lab called Mila MLM Mila.
The release of ChatGPT marked a groundbreaking milestone in the development of AI.
Simultaneously, it's triggered concerns about it's a few words becomes paragraphs of text.
The world's first AI rules to address these risks have recently been adopted in the EU, but the regulation debate is still ongoing.
So what are the emerging realities of AI risks and how should be regulated to avert potential crises?
Joining us to answer these questions is a man often referred to as the godfather of AI, Yoshua Bengio.
He's also the founder and scientific director of Mila, Quebec.
Hey, this machine learning math in the background. So linear algebra and partial derivatives.
Just calculus. Yeah. So, um, about a year ago, you first signed an open letter along with prominent experts expressing your concerns.
This all went nowhere because nobody cared. Okay. This year, you signed another open letter,
again restating your concerns about regulation and the risks to society, how things changed over the last year.
Incredibly.
Um, I don't think that the people who signed the first letter a year ago expected that there would be so much global discussion about AI risks.
Now, I think we still have a long way to go for more people and environments to understand those risks.
But the, um. Around the world are worried about is cybersecurity cyber attacks.
Right. That could be real. Have to realize that there's a rapid increase in the ability of AI systems in terms of programing abilities.
Right now, they're not as good as the best programmers, but we can see the trends.
It's crazy, right? Writing Python code. You know, two years, five years from now, they'll be better than the best programmers.

But the problem is that same tool is dual use.
It could be used by terrorist, by, uh, you know, maybe some rogue governments.
You're. Can become dangerous when the AI systems become worm GPT.
Yeah, you're right to use this system of rewards to train like a grizzly bear.
Uh, what companies are trying to do? Chat's pretty fascinating.
I'm sorry. I am not going on chat. Okay? That will be safe. Yeah.
Professor Bengio, what you described, you know, about the caged animal that breaks free or we still have a gap to human intelligence.
There's. Okay, now, he's basically, uh, you know, hypothesis, hypothesizing,
release some what human intelligence even means and what intelligence is strong incentive to do.
Okay. And then would I actually be motivated by itself to say, I'll be determining I'll kill people.
You know, in my mind, it has zero motivation to do that.
Uh, because humans were animals, after all. And even lions and cats and dogs fight.
We're not that much different from animals, okay? We fight for existence.
Survival reasons. I want your food, and I want your girlfriend. Order!
All right. That's actually where we fight. I want land grab.
I want all this land. Air has none of this intrinsic motivation.
At least not the AI that runs in a server. It has no reason to control you and run your life.
Okay? Could care, couldn't care less. So those kinds of risks are completely overblown.
They're pretty dumb, actually. So I want to go here though.
This is extremely cool. You should actually go and watch this afterwards.
And you have time, which is they have a whole series on AI that is actually what this is.
Okay. So you know, select somebody, the NHK, you know, once, but then I'll play just a tiny bit of this actually.
Very cool. Um, how do you ensure that all this benefits everybody?
Okay, I'll go first and will monitor the things. Okay. It's interesting to the Japanese television goes all the way to Canada and talks to him.
Hello and welcome to Deep Blue, coming to you once again from Montreal, Canada.
Okay, let's see what she asks. Or explain exactly what's happening in AI.
If not you, then who? I mean, is there a chance that we may never fully understand how AI works?
It is possible. Um. One angle that helps understand why it's difficult is that this is a little embarrassing, by the way.
Actually, because no scientist, no biologist might tell you something like,
I have no idea what the [INAUDIBLE] a cell does, but with some cellular processes, yes.
We don't understand all of biology. Okay. But the basics are there.
We know what DNA does when our DNA structure looks like. Okay, we know what it means to be alive and what diseases aged.
And actually very pathetic state, which is even the basic cell.
Right. Pretty poorly understood. Even just until a few years ago.
Deep learning for you in general was a mystery because papers like The Unreasonable
Effectiveness of Deep Learning are actually out in Google papers like that,
where practitioners that do the math tell you, I don't know how this work, but somehow this classifies to a 99% accuracy.
That's like really embarrassing. Okay. It's like a black box to us and we can figure it out.
That's why dumb transformers are a little bit like that.
Okay, I'm going to tell you a lot about Transformers today, but Transformers at the end and then at the end of the day, so to speak, they transform some words into some other words like queries and tensors.
You know, that word prediction. But then at the guts of it, they're doing dot products between keys, values and queries.
But beyond that, people don't know why does the grammar look so good or how come it gets so much right?
You know all that, right? It's still a mystery, even a Transformers mystery. Partly because we have not written good visualization tools.
Actually, in my opinion, it is possible to go and know exactly word by word wire transformers outputting the next word, next word, next word inside your brain should be possible. There are small attempts to actually go there, but for the bulk of us it remains a mystery.
And it's really sad. It should not be like that. We should all understand 100% of exactly how it works.
But anyway, that's what he's talking about. Okay, you asked the Turing Award winner.

Do you know exactly how it works? Well, not really understand what is going on in your brain.
It has 80 billion neurons. And even if we understood the principles by which those neurons, uh, do their computation and adapt.
So that allows you to learn, you may still not understand how they come up with your answers.
So of course, researchers are trying to study not just human brains,
but also what's going on in these really large, uh, neural nets, uh, artificial neural nets.
But maybe there is no answer there. But how do we trust AI?
Okay, that's, uh. I mean, you know, he's Yoshua Bengio, and I'm not Joshua Bengio, but I'll basically argue against him.
Okay. I'll just basically go up against him. So what he said is, like, very weird, actually.
He's trying to tell you that the human brain is so complicated, 3 billion neurons.
And then we don't understand how it works. And there are 1 trillion connections or something in the, you know, GPT before.
And then how do you expect it? How do you expect to know how it works?
Well, here's the huge difference okay. We cannot peer into our own brains okay.
And then record exactly what every neuron does. But we can do that with GPT.
We have every variable I showed you Lindsay, the other day. Okay? We have 100% understanding of how the pieces work.
And when the pieces work, there's something called emergence, which is something happens at a higher level, like traffic.
You know, you're sitting in traffic, but suddenly there's something called traffic jam.
So you are not a traffic jam, okay? And the traffic jam is not a thing by itself.
So individual cars cause it, but the jam is at a higher level.
We call that emergent phenomenon. Yeah, sure. So there might be something like that.
An alarm also. But we will be we should be able to understand how the basics work.
So it is not okay to blow it off by saying, well, it's like the brain, you know, it's not not at all like the brain.
So that's why it's called a cop out. Okay? Is coming up with an excuse for saying we are lazy to understand how it works.
Well, that's a stupid question. I'm trying to.
I'm going to move on this question. The languages are.
Different, but these are very cool. I'll tell you about robots, also robots.
And that's one of the research directions I want to deliver. So keep this, okay.
It's very cool. I thought it was possible. Um, yes.
So definitely, you know, I should say AI is big tech dominance is also that there's only a few handful of companies in the world,
basically Microsoft, Google and OpenAI, but to be exact, but control most of all the large language models.
Okay. So in that sense you have to have a monopoly, but there are other things that are happening in the world that are actually
going to break the monopoly so that we don't have to worry about that anyway.
So it's fascinating what he or he says. Want to say no.
Yeah. So that is the whole thing I just told you. Uh, try to make sure the benefits of AI are shared across the planet, across everyone.
Uh, we want to make sure that I. And we thought about large language models in Ethiopian.
What about large language models in Swahili that are. Or Telugu or something?
So that is how you make it equitable to the whole world. It can be always in English, beneficial for society.
So Chinese, you know, China is a pretty big country. So there are Chinese labs, but not necessarily in every language in the world.
So until that happens, you cannot call it equitable, right? In March of this year.
All right. So I want to move on this little cooldown. So I want you to watch it at some point.
What else. Yeah. So now you actually become like more practical.
Your homework, as you noticed, all four of them have to do with information retrieval.
Three are explicitly about rags. The fourth one, the Json one, is about input to a dragon.
So you can even do it on your own if you want. I want it to do crawling.
I want you to do crawling, but crawling. Not for tfidf, but crawling for just raw data.

Simply give me all the sentences, you know. I mean, we can basically question use that as a thing for answering questions.
So this is something we will definitely look at today. You know this already, but you have told you this so many times in here.
But one more time. All of, uh, rag works like this.
In fact, even Transformers work like this. But that's separate matter. In Rag.
What happens is. So you have a PDF file, and then you want a PDF file to be used to answer some question your query queries variable in homework.
For number four, you would initially take your PDF file and then take it sentence by sentence or even paragraph by paragraph,
and then run it through a neural network and then produce this one vector or an array of numbers called embedding, meaning one sentence can become an embedding. One, uh, paragraph like this can become an embedding.
One section even can become an embedding. Maybe not. The whole book can become an embedding is too big.
You get a chunk. It is called chunking.
So you chunk the initial data into all these embeddings, and the embeddings will all become embedded in this multi-dimensional space.
Dot that data. Then when your query also gets embedded, then the query vector will also hopefully be near the answer you're looking for.
So that is how it works. So you had a C plus plus book for your homework.
And the book had things about object orientation and polymorphism.
And if you ask what is polymorphism. That query called what is polymorphism will produce an embedding that will be very close to the chapter
that says polymorphism is a type of programing where you can just use the same function name for,
you know, multiple parameter list or multiple output types.
It goes and finds exactly what you're looking for without you going page by page are going in the index, right.
But that is done because your question look like the answer or the answer question.
Well so that's why you need embeddings okay. Then you can do all these cool things with embeddings.
But here what they're telling you is this is very cool.
By the way, this company called Run Galileo, they basically rank company.
So they are investing I want to play this video.
So they're basically highly invested in telling you how rank works and how you can choose the best rank model.
So that is why that large language models okay, let's watch this Precedented possibilities.
But going from a flashy demo to a production app isn't easy.
That is where most people get stuck. Drag fails use case.
How do you evaluate them? Outputs. How do you minimize hallucinations?
With Galileo, data science teams can go from demo to production in no time.
Let's say vehicle companies have jobs available. You should look at tuition and observability across the entire lamb app development lifecycle.
And so this product is called Lamb ecosystem product.
They did not make one mobile alarm. They didn't make a reranking library.
They didn't make an indexing algorithm. They're just simply giving you the platform.
You can evaluate what works and what doesn't work. You can pull your data and track your experiments to build powerful concepts.
Companies need this like a microscope to say what went wrong? Why did you answer me incorrectly and get alerts the moment this cool right off?
So it's going to peer into the algorithm to see like what it does.
Okay, well you're wrong but not understand how it works in moderate risk evaluations.
It started in minutes wherever you work. Okay. So Galileo is very cool, but I want to go back to the thing that I had here right.
Or there with the hallucination Index. Yes. You can read about all this. These are various metrics.
In other words, if you are a real company, you want to make a chat agent.
There are millions of people in the world would use.
You need to take multiple lambs and evaluate each lamb and give you the score, and maybe add all the scores and then

choose to go with the best score.
That's that's what it is. Okay, cool. So it's very useful.
Yeah these are different kind of embeddings. So the word embeddings in general will take your input like sentence paragraph I told you,
or even music or even images or even video and turn them into ultimately like a single Python numerical array like numpy.
But then how do you do that? There are so many different ways of doing it. They're all different neural networks.
So each neuron network is called an embedding model. So you pick the proper embedding model.
And then you pass all your input data through it. And you will get embeddings. And you're basically also through the exact same embedding model.
Then you can compare them okay cool. So it's a pretty informative article.
I won't read through all of them. You will know so much about how all this happens here.
That is how the embedding model is trained. And once the embedding model works correctly, then you can start embedding lots of data into it.
And yeah how do you measure the embedding performance. Is super cool.
You know and Galileo has an API. So you can even try this with your own LMS.
I simply throw you in kicking and screaming, okay.
I don't know how to swim. I don't want to get in the water. Oh, come on, you'll figure it out. That is what the homework was about.
Yeah. Some of you had problems with the shell script and, you know Python, right?
But you can solve all of that. It doesn't matter. The last homework, the number four lightning that I.
That is the magic book that tells you the problems you had with the other three questions are all solvable.
Like, make them all run on the cloud. Make a container. So as soon as you click on it in five seconds, it starts running.
Okay. So we have solutions. Therefore, I hope you didn't get angry that you got stuck with some no Mac platform where your thing didn't run.
It was a temporary okay. Okay, so this is great. I'll just leave it at that.
So we can evaluate rags and, you know, different drug, uh, platforms and then pick the very best one.
So this one is multimodal rag. So in multimodal rag, you're doing again some external augmentation to answer the questions.
But external augmentation can consist of a bunch of images and also a bunch of words together.
And so then you can retrieve images. You can do image search by typing in, you know, or vice versa.
Give it an image and say, what word would you use to label this? So you can do like more things than before.
Automatic labeling of images. We call all this loosely multimodal rag.
So very cool. So because initially all the rag was all text based, joint embedding is what that is called.
You jointly embed two different modalities meaning text and images, text and video, audio and video, audio, text and images.
Why not combine all of them and do a joint embedding so that a bunch of text and a video to go within will become one vector, so to speak.
So that way you can, you know, combine them and embed them okay. And then the query would then also be very similar.
You can then do a combined search. So like here. Great.
So you can you can embed so many different things when music videos.
Yeah. So all this right a train pulls into a busy station and then it'll go and actually bring to a video of a train pulling into a police station.
I mean, if you're in Hollywood, right, and making movies, this is basically a dream for you because that is what directors want.
They want to quickly just visualize shots in their head.
But then usually assistants are there to get all the stuff for them, but now they can do it themselves.
A sports broadcaster can even start typing the amazing NFL quarterback that scored a, you know, goal in the last five seconds or the game.
[INAUDIBLE] go and pull the clip from actual NFL footage, right.
These things are basically currently impossible. We use things called digital asset management.
We call them dam. So dams are how radio indexed sports videos are indexed, but they're all keyword based.

So unless you type the proper keywords good luck. But here it's not based on keywords.
Please remember that over and over again. Okay. Like here right.
Image plus audio. Wow, that is crazy. Penguin calls silly play.
In other words, if you play penguin calls, the system knows like bird.
Bird sounds. It identifies. Wow. You know, he sure is showing me how penguins, you know, talk and go and pull up penguins.
Video. That's pretty cool. Wow. Okay. Whale song.
You play like a whale song. Audio. It'll go on. Bring up whales underwater.
All right. It tells you exactly how to do it. Okay. So we we. It's pretty nice.
Um, one of the homework sessions was based on we it so we we it has a multimodal API.
So you can try this to yourself. You know, here's what I would do for you.
Try what they're giving you.
But don't go do it in lightning dot AI and make a brand new studio and make a brand new template is called publish it with your name on it,
and other people will start using it and start leaving your comments. That is how you get established in this field, okay?
You don't have to do it in your Python notebook. Yo lightning! The lightning people, by the way, are super friendly.
You can do this. Lightning, lightning people. By the way, I want to actually, you know, work more with USC this fall.
They might come and give a talk about how, you know the setup. There are PyTorch lightning if you're familiar with that.
Okay. That's where the word lightning comes from. Let's see. Your hands are on LinkedIn and answers questions.
I become friends with both the business development people, the very small group in San Francisco.
They really want to help us. So you can then get in there and actually start becoming an author of those templates.
Great. So yeah, go on GitHub and start building and then put it up on lightning.
What about this one? Wow. Okay. So someday if you think you know, I want I want to make my own, um, I learn from scratch for no reason.
Okay? I just want to make my own, and I'm sure it happens. So you have and record parties, you know, code, right?
For Jupiter to event. But the next thing you need is data, right? I need actual words like tokens.
So these papers give you a massive amounts of data for free. Whoa.
So this one is called refined website for sale. And Falcon is one of the elements made with many trillion tokens.
Right? Trillions of tokens. So you can read this paper and I show you the actual data set in a minute.
So amazing. So it's very interesting that people write these kinds of papers, but massive data.
This paper is mostly about data, but tokens really used as inputs for their lives.
But then this next one, this one here that says, uh, one.
Yeah, fine web scenario can actually go on, uh, find the hugging face and get the fine web data set.
This is the one with many trillions of models. So tokens actually crazy, right?
Yeah. It might be a model card that you can actually read about the files.
Ultimately what you want to generate. You know, the it's not this obviously it might be in here.
Maybe you say any one of these is a massively large files 2.1 gig, 2.1 gigs.
Just imagine getting all of them. You're going to get like a thousand gig total, right?
That might be like, you know, one terabyte or something. But all of those can be data for actually training your model.
Amazing, right? Wow. You can actually download it a two gig file containing pure text.
Okay. Well, it means sites like these basically give away data.
So someday you can train your own LMS if you are in university and like this, right?
You are doing research on LMS. You need to know exactly what you put in so you can query it to see where the answers came from.
OpenAI will not give you access to the training data set. Nobody even knows what they use.
So you need things like fine. Okay. Okay, we're almost here.
This is one of the most amazing meaning, one of the most, uh, comprehensive articles about rag.
So one understand. So rag is actually one of the key magic ingredients in all of this.
Um, we'll keep using rag for a long time, because rag truly works.
Look how big this page is. So spend time after this course and 100% understand it.
There's nothing that is mathematically very difficult. There's almost no matter at all.
Okay? But it tells you that a database. Graph database. I told you, rag can rag just means knowledge coming from an external source.

The source can be a vector database, like all your homeworks, but there can be a SQL bunch of tables, relational tables.
You can run SQL on average a column and use that to answer your query.
You know, saying anything is possible really. And here the compare vector databases with graphs right.
Yeah. But then your question your question will also become remember I told you and produce the nearest matches, the top three top K chunks that answer your question and then rank them somehow.
And then they will then spit it back to you. So we're still at the very top.
Write it. And today I'm going to tell you the difference between rag and fine tuning anyway.
So cool. I mean, it just goes on and on and on. So in your homework, you had basically all these.
I just know a few minutes ago I told you this PDF file.
Break it up a little pieces called chunks, an embedded chunk, one chunk, one vector and indexed them all.
The first search and then your you your query will also get embedded and then go in the exact same bunch of embeddings,
and then pull out the top embeddings that match your query and give it back to them and tell generate natural response for you.
Is very easy. It means you didn't answer your question. And that's what you want.
So again to give you a code you can run I mean it's it's one of the most comprehensive, uh, pages that I've ever seen.
Snake that people make stuff like this. These are what will come up over and over and over.
See, in the beginning of this course, we want the context in a recall, right? Things like that.
Precision and recall. That's what this is. Likewise cosine similarities in there somewhere.
The basics never went away I like right there. Right there. Okay.
And then multimodal ragged editorial, just like Lama Salama is one of the open source libraries from from meta.
There's also lava which stands for vision. So now it's an image transform image a la la la la mano la okay.
See text and images together they become embedded. Quote.
Actually they became embedded here. And then the query will then pick both of them and then answer it for you.
This is also cool. You can read. Rag has problems.
I told you in your homework for the question you might ask for might not have the right answer, but that's not the aim of this exercise.
It is. Just do it so you can submit slightly incorrect answers.
But in the real world, that won't fly. Your company cannot answer the customer's questions wrong.
We have ways to fix it. Okay, so I will even tell you about rag 2.0.
In fact, it's right here. So there's a big difference between rag 2.0 and the current generation of rag that we call rag 1.0.
So rag 1.0 is not that great. Just to tell you. So rag 2.0 actually is not any better.
Way better. Also, what might be shocking for you is after rag 2.0 in 5 years, do you think there might be a rag 3.0?
But somebody might tell you in five years the whole notion of rag would be obsolete.
You've got to be prepared for that. Okay. So rag is a temporary thing that came and went all of last year.
People said prompt engineering is the job of the future. $1 million per year.
This year, nobody talks about prompt engineering because there's no such thing. Okay.
We're all prompt engineers. So some of these things are basically a lot of hype.
You cannot, you know, just be obsessed and go down that rabbit hole.
So rags actually will disappear okay. The whole thing will disappear. I'll even tell you why.
But before that, though, Amanda, AI is not just about Rag, it is a general site that explains almost everything about AI.
It is super cool. Serious. So you want to learn about recommendation system.
Then there's a whole big. That is what's called deep dive.
You know that truly. I mean, look at this. Each is a whole page, okay. So we have no time to go through all of them.
But really, if you spend time, you'll you'll know so much about basically everything.
Really. So please, please, please read. Become educated.
Get those jobs. Okay. Neat.
Uh, so that's a man that I. And what else? I wanted to show you rag 2.0, yeah?
You know, the so-called rag 1.0, which is currently a running on your homework, has a very, very interesting problem.
A problem is that the core lamb that you're told not to answer, your query might not really have any context,
any background, anything to combine the retrieval with the two different things almost.
So you have a core lamb here, right? Quite a meter. And the lamb might know something about your query query is not that amazing.

Remember it could have told you what polymorphism okay. But you told it specifically.

Go here in your PDF file and then come back with what polymorphism means.

I mean it all the lamb. Now give me the answer back. That means that is called retrieve context.

The query has, you know, words it needs to answer, right. But the extra words to answer your query, basically the answer itself, we call it context.

So the context is pretty much what rag fetches. Every rag fetches context from those external sources.

But the lamb also is able to generate context, in other words, the words that it produces,

which all of this the result of as predecessor is context now,

but because these have no connection with each other, the context might mismatch with each other.

And so that's a bad thing. That means you are entirely stuck only using rag output.

You can never try to combine with actual output. If you do, there might be more weird hallucination, but that problem is not unfixable.

You can actually do one integrated retrieval across that lamb and drag external memory, and that might have uniform context.

That idea is called rag to piano and rag to point out the difference is you're trying to make a combined context with the actual LM.

Oh, no. Oh. Uh huh.

All right. If not, it doesn't matter.

I was going to show you this article, but. Okay, try this one.

Yeah. See in here. Right. It tells you what is like 2.0.

A typical rag uses of frozen off the shelf model for embeddings, which is the alarm, and then as a separate vector database for retrieval,

and then finally even take the retrieval output and make the results that humans can understand.

That is a black box. How it turns the embedding.

The context results into natural language. In your homework you can tell how does it make sense as a black box, right?

But you have this Frankenstein's monster, which is a bunch of individual pieces, three different pieces.

One is the text generation piece and one Estella own context that you can never use, and then actual context from three different things.

What if we integrate all of them? Then that's a much better read.

That is basically what read 2.0. I see this here in rag 1.0.

They're all different, right? But then in rag 2.0 end to end, you know, all of it.

You know, it's a combined rag across like all the components.

So then you're able to get, you know, much better results. That's all. Okay.

Then there's all these benchmarks that people use to measure how Iraq knows better than Iraq.

So we've already come to that point of talking about Iraq 2.0. Great.

When you hardly thought that you knew about Iraq and 1.0, but now suddenly Iraq two point now and there's going to be a fall of the.

So we're not to worry about it. One more one.

Every company in in, in in the end wants to build trust where they introduce our new solution for the customer condition.

It is designed to revolutionize the world of retrieval, augmented generation and help you implement both simple and complex use cases.

Rack shortfall retrieval. Augmented generation is a super exciting method with tons of use cases.

Most importantly, it lets an LM efficiently read your company's documents.

For example, to be able to submit a company's documents within seconds, you can see it, right?

So we can retrieve internal knowledge and features into a language model and use it for very specific tasks,

such as I should have said, external knowledge, which is gathering information or rephrasing complicated texts.

Okay, so this enhanced capabilities um, and to implement actually here though is extremely important.

You know, against say a Bank of America customers want extremely specific information about the kind of loans that the bank gives out.

If you have some kind of rules, you know, a company that's a big supermarket,

your customers want to know exactly what is in certain food items that they consume and got sick.

They want to know the ingredients. Okay, that is not the time for them to make beer get sued.

That is the reason why they will never deploy. Standard algorithms are pretty horrible actually.

You know, they they hurt the business. So you need, first of all,
for the area to even know what the customer wants and then answer them with a high quality answer or just say, I don't know.
It's better to say I don't know than to make crap up.
So that is basically what drug users obviously want are going to ask for something and to implement strategies.
That's according to the user's intent. So that's the the external memory.
Okay. I'm going to keep going. It's a nice thing you're going to watch out for this company.
So companies right have all these PDFs. So you know I told you to use any PDF you want.
But when you join any company they'll have all kinds of employee handbooks and, you know, documents and whatever.
Right. So then that is what that the rag has to, you know, used to answer.
I mean, look at so important. So suppose you ask a company what kind of data do you collect on those guys?
It needs to exactly answer that not make up anything. Right. This also how long do you keep my data?
I want to know. Then it should come and tell you it's 3 to 20 years. Okay. Wow.
So drugs are collecting zero surance company dealing with a multitude of complicated insurance policies.
In this case, we can use cognition and its enhanced capabilities to help customers and their traveling
incidents by quickly finding the right piece of information that they are asking about.
And we can look, you know, in a law firm. There are thousands of cases that they know basically argued in court over the years.
And all of those briefs are available in order to pass judgments as PDF files.
No lawyer. You won't have to have a flash drive full of 1000 briefs.
Has a time, energy, you know, whatever, right?
To look through all of them and there are no hurry out rushing the one to know what happened ten years ago.
Rag can actually be used to pull exactly what they want. It's like a magic.
Okay, in in law firms to hire junior lawyers just to do that.
And junior lawyers are bored because they didn't get hired to read like old briefs.
Okay. They want to go and argue cases in court. This might be a perfect use for it.
Likewise, doctors can even the memories feeling they're getting old.
They want to treat a patient who is in front of them, but then it reminds them of a patient they had 20 years ago.
Can you get that person's medical history? Good luck for the doctor to search through all that stuff, right?
But you can actually do it leverages the power of language okay. So so-called reliability the most have which.
All right. Set up the project. The slick so-called once again you know this company current actually give you some code.
You can write anything up for you open source to run it all.
Really. What concerns do they have and what information are they seeking?
Okay. So let's say that you notice a nice thing about.
Rag. Okay, here, you message generation.
You know this way. Cool cross encoder to actually make the retrieval look more efficient.
So many things. Okay, so currently is a company at the Mighty One have jobs.
You know, if you go and look carefully. Right. Careers.
So if you think that you know you're good and you I want to work with all of this, just look at it.
You know, solution architect. It doesn't say senior solution architect.
Select like this tools for data centric AI.
So again things like lightning AI and how you actually package it and how many other people use it would be your portfolio.
I can actually tell them, look, I know how to build this. By the way, lightning is so cool.
Every time I go and see there are new templates I prefer, but I tried them.
By the way. I tried like more than a dozen. Okay. And I'm going to try them all over the break.
I mean, I'm going to try and add some of my own. I want to make 3D models.
I actually want to make meshes and then contribute them back to like lightning.
All right. So drag 2.0. But I'll tell you more about two point.
Also in the actual slides I have uh one more thing.
Yeah. Oh this one is also called Speaking of Lightning.
So lightning has their own streaming geospatial data format.
So in this case they wrote like a studio to actually showcase how that works.

That is pretty neat right? Geotech files. So JT files are basically, uh, Json files that have location information.
You can go stand in your atomic region and get the latitude longitude and then write them down two numbers,
the two column format, a third column time iteration, some kind of description, right?
If you make a Json file with thousands of locations throughout USC, that will be easier to file.
So just want to read them and make it streamable. So when you have millions of data points, then you can actually stream like them piece by piece.
That is what this is. See here. Right. All right. So location information with something else that is non location.
For example all the locations of Starbucks or I will place in L.A. where crime happened last year.
You know sort of spatial data with non spatial data. So for large data set you cannot see that right.
You cannot copy them. So you need to stream them. And so that is exactly what this particular studio does is extremely cool.
And again there it is. And then within a minute I am not going to click on it now.
But you can see right. You can say open in studio like right there.
As soon as I click on that you can run this. So this combines some things you already know which is spatial data retrieval.
And then now you can do a lightning streaming format.
You need to learn to combine like one thing with another thing and then do something better than both of them.
Hey, cool.
So now I only have that one slide from, uh, Miscellaneous Topics part two, and then a small set of topics for today, which is actually part three.
I'm going to go to part two and tell you about geospatial data, how it is going to make search even nicer.
This beautiful okay, I'll show you how to build this. So that is valuable knowledge.
Say you join this particular, uh, division within Google location based information.
They might ask you, how did we put this data set together?
Many people might not know. I'll show you a quick calendar.
Okay. All right. So then this is all like neat things we did.
So that's the whole drag idea. Can use rag you know.
And then you can use agents to actually basically work together. And then we call that an external operating system.
And so the creation of a operating system named entity recognition can be done automatically.
And then to go with the topic modeling, you can basically classified documents, you know automatically.
And then you can make knowledge graphs that are very specific nodes and wires. And then use that knowledge graph can be used here by the way.
So one more way to do Rag is through a knowledge graph you know. Okay. Like neo it can be used.
And then recommendation engines are like extremely you know I based these days I talked
about monoliths and how they use a pretty cool a cuckoo um hash which is two hash tables,
data structure to actually bouncing on collisions back and forth. But wonderful.
And then Twitter's so then this just simply going into very specific details.
Uh, there are about 50 or so engineering blogs in my case 585 page.
Actually I will link to all of them. And maybe, you know, this weekend I'll actually put them for you guys as well.
50 different engineering blogs, Twitter, discord x meta, Microsoft, okay, uh, Hugging face, Grubhub, you name it.
Okay. They all have engineering blogs in those blogs that talk in great detail about the architecture, the so-called systems, Dag, the pipeline.
The more you know, the more you will connect between like them and go out.
And then there are hundreds of companies, but they're doing ten different things over there.
Once you get to that stage, you can confidently ace any interview.
You know, you can even talk about things that the innovators not not talk about.
Stack share. There is a nice site called Stack Share, stack Shared IO and stack IO.
All these top companies are named and even more they, um, have these diagrams.
It's very colorful diagrams. So I'm going to show you some where I'm looking for browse stacks.
Thank. Text. Text. Yeah.
See this one. See right here. Right. Uber for instance.
So what us over. Right over is pretty amazing I want to work for over. The first thing you need to know is not all of these are cool.

But what products does Uber actually use. They use react maybe the US and react okay.
In other words, go learn react. You know they use PostgreSQL for some relational tables, you know okay, if not go download it.
You can install, download learn all of them. Use the cloud obviously.
Right. And then they use this, you know, js API call backbone.
That's pretty cool. It's like a, you know, a way to augment vanilla JavaScript code like jQuery.
All that back in the day. They use park. So like in-memory fast processing they use Hadoop for big data.
You should know at least each 1 in 1 line.
What the heck all these are? You should be able tell them. Even better, you should have done one project on each one of them.
You know, that's by the way, top 10 or 20 tools that all companies use.
So it's not even hundreds once you master it. And there's even articles, by the way, that tell you how there's discretion some way.
If you read about how, uh, these actually put together because it doesn't tell you what the system diagram is right.
For that, I would use Alex. Sure. I told you so.
Alex shoe is a pretty cool guy, you know, you can follow him on LinkedIn, CSS, amazing systems design stuff.
You look right here. Right. So these books are free. You can just get it. Okay.
That the book that in combination with these diagrams that I'm showing you would then basically tell you everything you need to know.
Byte. Byte is the name of his site is wildly.
See that vector? So pretty.
Like I told you in share, they might name all these things one at a time individually, but it shows you how it's connected.
Next step is you should know why they're connected like that. Next up, if somebody asks you what is the alternative?
Like what can I replace? You should be able to tell somebody there just piece by piece by piece.
You can learn okay, this is great, right? It even tells you why, you know, all these little you know, all these from data structure, cybersecurity.
One word. Right. But now time to put them back together okay.
Anyway. So it's even explain SQL by the way this is one of my favorite articles.
It tells you how SQL engine works. All right. So that is recommendation engines.
You can learn a lot I talked about clustering. These are three different but you know almost identical clustering algorithms.
In clustering you cluster these embeddings in multidimensional space.
You know. But sometimes there's a need to combine two dimensions and reduce it to one dimension.
And you can use any of these algorithms. And then for visualization purposes and only for visualization purposes,
you can keep on combining to work toward a time, to a time to time, until you end up with a net final result.
Only two dimensions. You might have ten dimensional drag context vectors,
but nobody can know in ten dimensions what the clusters look like and would be a regress producing wrong output like wow,
what the heck is wrong in doing that? You can go to TSA and go from ten dimensions all the way down to 2 in 2.
You can take all these vectors that you put in right and color them somehow and actually in 2D, look at the coloring.
In other words, you can visualize clusters in 2D. Next plots of paint that somebody dropped, multiple splats of paint, a drop in all the paints,
just all that can tell you, oh my God, why is one cluster so much bigger than the other?
I thought I input Q even amounts of data. There's something going wrong with my embedding layer.
You go look at it so you can very easily do t-SNE. I told you it's called stochastic neighbor embedding.
So look at this t sne clustering.
The big value for us lies in the visualization I told you, because once you get down to 2D, that's what I mean, right?
Maybe you expect all clusters to be evenly distributed. That cluster is such, right?
What the heck? Why is I don't get it right? My embedding is not working properly or in here.
You expected all the embeddings to be separated. Why are they basically combining all over the place?
So it's a very fun way to actually, you know, debug things, right? That is why you can use it for.
Okay, I'll show you. Pretty soon you'll be blown away in charge of pretty.
The clustering happens in about 50,000 dimensions because each dimension is for one word.

In the English language, we use about 50,000 words most commonly.
So it shows the dimension.
Okay, so we have no idea of all the trillions of tokens that OpenAI trained, you know, GPT four on what to look like at 50,000 different dimensions.
But imagine slamming it all down to two dimensions.
So all your clusters are now on 2D. And look at it. Okay. It's pretty fascinating then you know.
Oh my god, OMG this actually works.
Okay, so then searches again the same thing such as such as such and such such search will be with us forever and ever and ever.
Discord has many billions of images, messages back and forth every day.
So people go and search through all channels, right? Heart attack. Pull it up.
They use term frequency, inverse document frequency. So there's nothing wrong with the basics at all.
You never completely throw it away and say, I want to do rag and context in embedding.
There is no need if there's a need for doing tried and true old fashioned technology, stick with it.
So that's what this tells you. Great. So I told you like all kinds of such as Reddit.
On the other hand there's vector search. So you can obviously do a combination of them.
It's not this or the okay. And finally, this is the new thing I want to tell you.
Okay, so as we speak, if I go and type, you know, anything, right?
Chinese restaurant near the US campus. Starbucks near me.
This is the state of the art today. Otherwise, Google Restaurant is looking better, right?
So as we speak in real time, this is what we get. So we get a map with all the nice locations.
Not bad, you know, a little helpful. I tell you where you are and go and find it.
Right. But what is the next step?
What if I want to stand right here with my phone and my laptop, and actually want to go into the Starbucks to figure out an expression?
I actually want to go inside and one look around and maybe decide, wow, this is not a place where I can go and do work because it's very narrow.
There's only four places to sit. I'll find a different Starbucks. How can I do that now?
Right? Yeah. There are photographs that people upload, but photos are 2D.
You want you want actual spatial data. I want to be able to walk through every classroom at USC to actually feel get a feel for it.
Right. How can we do that? Incredibly, this already exists.
So Google made a thing or two three years ago called Immersive View.
But I'm going to show you more than what is in this article. Immersive view.
That's the following. This looks like video, right? First of all, it's not video and it's a completely rendered image.
It's a generated image. So rendered image that is coming from point clouds.
Each thing that you see there is a little dot, dot, dot. You know oh my God that little point cloud.
And the point cloud is that each point obviously has a color in opacity.
How did they get to Point Cloud? I want to show it to you. And what can you do with the point cloud?
What you can do with the point cloud is fly through it any orientation, like a quadcopter yourself,
because there's no way in [INAUDIBLE] that Google would have flown every possible path through all of their own little space in the world, right?
And generated videos that didn't do that. They only used a very fixed path that is your input data.
But then the rendering, the AI, it's called neural radiance field Nerf rendering can produce any new pass that the input was did not do.
In other words. Look here. Supposing you have the globe.
Got the earth like a 3D volume. Okay. Imagine you have an object here.
Some kind of cool object like that. Object?
If I take a picture of the object from this view, if I look down, then I'm going to see this top like this.
Right? If I get a picture from this view, that's a different image.
If I get a picture from a 3D view for three of you, it's a sphere.
So I can basically go in 3D and take like many pictures, right? That is all input data, but those are discrete photographs in a folder.

Ten photographs of that object that I took, except I know where the camera angles are.
I know the location of the camera in a sphere. Also the orientation of the camera.
That's my input data. So my input data would be for each camera location and orientation.
What does a photograph look like of the object? In other words, I got my little chocolate box right.
I can take one picture this way. Second picture this way. Third picture, this picture all of it.
And use I. This way the reconstruction happens to basically learn what camera angle produces, what image.
And now we can reverse it. We can generate new images by giving it a new camera angle from which you did not take the photograph.
Then what it will do is find nearest similarity search.
It will find actual camera angles from which it took the picture closest to your query vector,
and then use these images right to synthesize a novel view.
That is what is doing okay. So it is beautiful.
So like that. Wow. Understand the vibe of a place before you go.
So the raw light, our point cloud data was all the false color, the red green dots.
But then it also captures color information I told you. Then they can turn all that into what looks like a live video.
Okay, I'll show you that. I'll show you how they made it. On the one hand, they use actual street view.
They actually generate street view level data. The standard Google car that you see with the camera.
They also use planes to use airplanes with cameras looking down.
Those are very specific kind of cameras. Those cameras are a little like this.
So imagine there's a plane. There's a plane, right?
It's a little plane, but the plane has camera, and a camera has so many lenses that all look like at the same point,
and they're almost trying to produce parallax. In other words, I look at all this way.
Look at the world this way. Eyes are not parallel, by the way. It's actually slightly curved.
Okay. So between the two eyes, we can get what is called disparity. Each image looks different.
We can get 3D from that. That is what the camera systems do. So they're flying planes like all over the world and actually generating data.
That is how they can then combine all the data set. AI seamlessly blends all of that.
So it blends a view from all the from the top to street view and down into the restaurant.
And then the tables of people are sitting and eating. I want to show that. You okay? So this is, by the way, today.
And like I told you what is coming. You can even make, by the way, video games with this because Google gives all this data away for free.
So as programmers, we can have access to all of them. I want to go on YouTube.
I actually say YouTube YouTube. Immersive use.
Immersive view. Google. I'll show you should there.
And then I show you how they actually constructed. Okay. Okay.
So let's try this one. Just happened so fast.
These are pretty short. So I'm going to basically like let them play okay. So you can actually run out of a budget okay.
Watch this. We're also bringing new capabilities into maps.
I'm going to turn the lights off. Why not.
Mapping machine learning using billions of aerial and street level images to create a new high fidelity representation of updates.
This breakthrough helps. Okay, coming together to power a new experience and maps called Immersive View.
It allows you to explore a place like never before. Let's go to take.
Let's go to London and take a look. What a beautiful city. Say you're planning to visit Westminster with your family.
You can get into this immersive. You say some that's on your phone and you can pan around the sights.
Here's Westminster Abbey.
And if you're thinking of heading to see Big Ben, you can check if there's traffic, how busy it is, and even see the weather forecast.
It's London, so I'm guessing it's rain. Now, if you're looking to grab a bite during your visit, you can check out restaurants nearby.
Okay, that's the cool part. Look, we can actually go inside. That's crazy.
So classrooms, museums, hilltops. You can go anywhere in the world.
What's amazing is that this isn't a drone flying in the restroom.

We use neural rendering to create the experience from images alone.
And Google Cloud Immersive Stream allows this experience to run on any smartphone.
This feature will start rolling out in Google Maps for select cities globally this year.
But look at this though. Um. Yeah.
This one. These 4.5 minutes are extremely important.
Okay. I could just use a 2D Google Maps, or I could tap here and go into Immersive View and get a 3D view of not just the Ferry Building,
but everything around it, including the water, palm trees, and even the seagulls flying by.
I can also see what the weather and traffic is like. Google creates these realistic digital models by combining using cameras like this.
You can see and 3D airplanes, and we're about to go check out those cameras and learn more about how Google created Immersive you.
So let's go check it out. Immersive you is Google maps is the latest flex.
Instead of seeing just a ground level image of a building or a landmark like you do in Street View,
Immersive View features these three dimensional, hyper realistic previews of your journey.
So you're going to Google Maps. Let's say you want to walk from the Ferry Building to the Palace of Fine Arts.
You put in the Palace of Fine Arts is your destiny.
That's the group, the target or the thumbnail up here, an animated thumbnail you tap on that just like you to look as a seamless flight through.
You can zoom in from the sky all the way down to ground view.
You're interested in street view.
Instead of seeing a red line to symbolize traffic, you'll see a rendering of cars backed up to really visualize what you'll encounter.
And even though it looks lifelike, these aren't real time images. It's purely simulated.
A rerendering of so many people have, you know, spatial difficulties in like, orienting themselves.
So when you turn around somewhere, right, Google Maps says, you know, go straight, you don't know what straight is.
Okay? So stuff like this can actually help, you know, spatially challenged people a lot capture.
It's not there's no live camera happening. The idea is to make it look real.
It's being rendered, which was called neural radiance field modeling, Nerf rendering.
I saw the cameras that power Google Maps versus not even the state of the art that's actually obsolete.
I'll tell you about something else. Construction Splatting images captured for Street View with cameras like this,
and overlaying them with 3D footage taken by this camera that's attached to a plane.
The result is these three dimensional bird's eye. That's why they got that inner landmark.
These cameras have a different geometry. They're all facing towards one another.
See that, Mr. Parallax? Okay, well,
of being in a nice ring with a common focal center so that we can create a nice spherical image where these actually have a different goal.
By creating parallax between the cameras, we can use that parallax to have a focal length in front of them.
All three converge the 3D. We have two eyes. Two is enough for by now it's called binocular vision disparity.
But why not have more? All the imagery and identify objects returned and nowhere else.
And road means when you piece it all together, you get something like that. That is the whole segment, anything.
So you can segment whatever they see buildings, streets, cars and you get could get other views from here.
You'll see. So what happened here? Right. You get something.
There's a little magical thing that's going on here, which is you see this is simply lidar point clouds.
But then the system is able to know that is a tree. These are people.
That's a building. That's a light pole. That is more AI. Because neural network neural networks are trained to recognize all those objects.
So an undifferentiated bunch of points, which is what the lighter cloud is that you got from the camera,
is now able to break the objects apart and then go inside buildings.
You can take the cloud if you want. So this is a new thing, right? Metta has a model called Sam.
This obviously Google's model Sam stands for segment anything model.
So Google the white segment anything model. Also Google YOLO collaboration eight YOLO version nine.
So in YOLO I can take a picture like this and then have the.

I put a bounding box around every student and say face to face, face and give me the bounding box as x, y and x max y max meaning location within the image, but it won't pull them out for you.
But then I can take the bounding boxes and the pixels as input data and give it to Sam and tell Sam pull out.
Just the students pull out just the chairs. So Sam was able to segment. Okay, so we can use Yolo and Sam together.
Anyway, that's what stuff like this is pretty amazing right there. When you piece it all together you get something like this.
You can actually it'll show you where the turns are. You'll be able to zoom in.
Wow. Um, and you get good news.
I mean this so you can download it, use it to street view tilt to see what it looks like.
At the distance, you can see the traffic. Let's go inside. Okay. Not a good time to go for a bike ride.
We have what's known here as the time slider. What's the sunset going to look like?
Google's cameras have clearly come a long way since Streetview launched in 2007.
If you can see, we have a long history of developing cameras here, and they have a museum of cameras.
Huh? And more compact over time and more portable. So this is the very first one developed for Street View.
So the history of state here so we can skip okay. Lighter than that 500 pound brake strap you started with.
But I personally wow. Look how heavy that is. Long. Oh my goodness.
My backpack to work. Google's latest Street View car which you pay people to actually walk around with that all day long and capture footage.
Wow. So pretty aerial camera. I used all this imagery together, which is volumetric rendering.
You need permission from FAA or basically to fly low and actually capture all that, but your Google afterwards you can get away with it.
And then the volumetric rendering. That's the whole neural radiance field I'm going to tell you about pretty soon okay.
And ray tracing. And do the CGI. That's pure 3D CGI versions of immersive.
You want to explore places like landmarks and parks, which is already available, but the trees look fake.
They look like watercolor trees, right? Because they don't care about each leaf or something.
That's a big one. Green walk soon. But it's okay, ma'am, you'll see. It's way better than what we have now.
Yes, when you launch Google Maps and with details like those seagulls I saw flying around the Ferry Building, it'll all feel a lot more realistic.
It's about the delight and the utility, not just the utility. But we want people to go to immersive you and go, wow, that's really interesting.
That, isn't it? Yeah, you had to walk sometimes, but it's also a little bit more to add to what had this to say.
There's a thing called CJ's.
So CJ's is a JavaScript library that is able to render these kinds of spatial tiles, actually 3D tiles, right?
Yeah. In other words, you can take data that Google gives you spatial data.
And you see some libraries actually make those photoreal renders that you saw to look at that shows you from and this goes back to 2014.
But see the GeoJSON at a level Json data, it's all latitude longitude but some map overlay that points of interest.
I want to skip all of those and tell you CJ's 3D tiles, because that's the more you, the more immediate, the more recent stuff.
Harder. One 2023 let me share my screen.
So by the way, AQ, QGis is a free jazz program called Quantum Jazz.
You can download it tonight is entirely free. And then you can try all this yourself.
Going. Hey, Amy, can you just give me another word?
Say you want to write your own search app. That is not Google Maps.
All right, cool to be on a bunch of restaurants, and you want people to actually go into the restaurants.
You can use a mile high. I'm going to kick off this. What if you made one for every USA classroom?
Wouldn't that be cool? Uh. Um. And we say the same liars.
It actually gives us an opportunity. Before the internet, there were no virtual campus tours of some kid who wants to tour the campus.
They would come to USC, right? But no, you are simply virtual.
If you sell a home Realtors, the state of the art is simply some immersive video you can click on and stuff, right?
They would go crazy with all of this, because your customer that is going to buy a home, right, can actually be in the rooms and walk around, go to the backyard.
That's crazy. So if you think of innovative use cases like that.

Okay, I'm going to move on though. That is the new part. The whole 3D tiles to watch this.
So, uh, focusing on the cesium 3D tiles.
It's a standardized format. It's the community standard.
More detail. The closer you get to it. That's why it's called that.
It's a hierarchy of tile information. They say that. In other words, at some level, the entire thing is one big tile.
And then you break it down into regions. It's called region three, but it's region train 3D.
And then you can break it down all the way down to individual level segmentation. The whole thing is a hierarchy.
Then if you go in this part of the building, it will first load this volume and then load even smaller volumes.
Assuming more and more you can delete all this data. It's basically fast way of doing research.
Okay. So again one way of indexing 3D data. Um, okay.
But you can build all this in QGis. And how cool is that you initially get I want to point out.
Yep. Um, when I get the data set has that sort of coverage.
So suddenly I'm going to just Google something different for you.
I'm going to say Google, Google photorealistic 3D tiles, photorealistic raters,
because these are the ones that are basically drag and drop into a Google map.
It'll actually work. Okay. Yeah. What about this one? Uh, 2023 again, I'll show you the next one with a JavaScript API so you can do this yourself.
Okay. Good morning, good day, good afternoon, evening, wherever you may be tuning in.
My name is Travis McPhail, and I'm the engineering lead for visualization within Google Maps platform.
I'm joined by Dan Bailey, product manager for imagery on Google Maps platform.
Okay, let's check that out. I have long desired the ability to power immersive experiences with the same 3D data source.
That's cool. Let's flip it. Google Earth has pioneered 3D photorealistic experiences for the past 18 years,
powering use cases from government, real estate, city building, and even data visualization.
But as a component of 3D experiences like check it out.
Well, it's time to showcase the components you can leverage today.
That's what you do. You actually make a computer graphics mesh like a character like character,
and it has texture maps and geometry, and you can render from any viewpoint.
That's what they call a 3D photorealistic. You can now access Google Maps platforms real well with one on the world.
So whatever's in the picture, sure, you can do it yourself. Thanks, Travis.
Hi everyone. I'm Dan Bailey, product manager for imagery on Google Maps platform.
It can cost a lot of time and money to implement your own real world 3D model.
Which gen immersive experiences.
Okay, starts here without having to have an API in so many languages JavaScript, Python, etc. specifically designed for visualization.
Use cases at city to block scale. Our photorealistic 3D tiles offer a seamless 3D mesh model.
So the question that I was showing you. But now it's available to actually actually.
It is one of the world's most comprehensive 3D maps. Zoom out, zooming in over 2500 cities across 49 countries.
Because it's georeferenced alongside our other maps APIs, plug in like any address you want.
It'll go and load the tiles and then render it for you. Overlay other Google Maps platform data.
Very clever visual. All right, so it's a geospatial data standard.
Yeah. That's all you do. See. It's so easy. That URL just get like some JavaScript code import to modules.
And then you can do it I'll show it to you okay. Regular Europe. Let's talk about our system JS.
Over the past several months we've been live for you. So please learn system to the real world visuals for your local view.
Air, water or roads. Where you request postal address.
That's why there's a geo. That's called geocoding.
Geocoding means you take an address and turn it into. Today we're providing ready to use it for hotels across the U.S. and which.
Like the church right there. Right.
Because I know you like. Think for a second.
Oh yeah. Yeah. Good question. Is that like something? Huh?
You know, that's a great question. You know, this question in the hexagons is no overlap.
It's like a, uh, you know, uh, you know, basketball or something, right?

It's all stitched and tied together. But in these tiles, right, there are overlaps and overlaps.
Okay. This comes from a 2D data structure called a region tree or tree in our tree overlaps okay.
So simply our tree in 3D. Yeah. Cool, a good question.
Good. Are you? Got it. Okay, so I want to move on. I'll show you the JavaScript API.
Then we can do other things. Okay. Check it out. This one. So this is something you can actually do.
Is the IT painkiller remote monitoring and management. Remote access help desk for ring on.
Check this out. Good morning. This is my goal from inspired.
I'm going to jump to code. Okay. And so excited to, uh, share like a little, uh, video there a tool called cesium.
Uh, just it's all cesium. Cesium, cesium, cesium is wildly great because you can take Google's photo real tiles and go in that protocol.
Omniverse. So cesium formerly works.
And being Nvidia's immersive VR and 3D, so suddenly you're being photoreal Blackwell GPU rendered Nvidia in VR world that is called Omniverse.
Okay. Likewise, you can make a video game and actually have characters running around, and you can have cesium running inside unity.
So cesium is this amazing product that, you know, at some open source, some kind of engine.
Yeah. So it's cross-platform in that sense. Uh, go watch this or let's see how they, you know, bring it in.
There it is. Cool, cool cool. It's so simple.
C++ just import that one script file from some CDN somewhere.
Okay. That's all. Then start doing file stuff. And then when you run this, it will actually work.
Read docs link. You'll get pretty much close to this screen and you want to go down to the 3D tiles subsection.
Okay, let's keep going with this API in this end down here.
That's the link here. Again same thing right. Just import the JavaScript file.
Just has number and the init function data.
Oh look it's going to run to. Great. And. Under an hour.
And so the magic of all of this is you run that code, then what happens?
Okay, so this code is real, but you can run this in chat, flutter, react, and make actual apps out of it while you get that, uh, if you're interested.
So this one is Google technology. Whoa.
That's pretty neat little virtual play that okay.
See our system? All that is purely magical. So I'm going to tell you that is really how I think, uh, location search is likely to evolve.
Okay. Next I want to tell you this notion of Nerf rendering.
Right. Well, leave the lights off. Okay. So one way you can synthesize new views is to be able to represent as a ray.
You march along this ray and then encoders and pixels. Right.
You build a model based on exactly what Ray. You know, how many steps you made and what pixels you get.
That is called neural radiance field nerf. You're modeling the radiance.
You're modeling the roughly the brightness and color of different objects as seen from different viewpoints.
Okay. So I'm going to run that for you, but I show you something more recent than this.
Also for now, neural radiance field. This is a very big deal when it first came out a few years ago.
Let's play this. Okay. Actually, that one is regularizing.
Hmm. Okay. Um, I might not even show you all of this.
I meant to just say Nerf rendering, just the basics, okay?
You don't need to optimize it yet, because then you can actually get what I'm saying to you in terms of multiple views.
One. Okay. Watch this. We present neural radiance field or nerf a new method.
So none of this is video. In the sense there are individual photographs and nobody made the video camera go like this.
There's no video camera. It took still pictures, and the continuous, seamless view that you get is all interpolated.
You will see a little bit about how it works. Okay, it achieves state of the art results for view synthesis.
It is cool. See that those are the cameras I told you about, which in this visualization is a Lego bulldozer.
So the same the same little Lego object is being visualized, being imaged, actual photographs from all these camera positions and also orientation.
And then the data would then become the photograph and the location, likewise a different location, different photographs.

Then you combine all of them and make a neural network learn a location of the scene rather see
valued function that's called re marching continuous five d coordinate consisting of a location.
And so that is the training data make millions of rows like that.
The scene representing camera image camera image camera image that takes and that's a table of data.
Make a neural network learn that the corresponding volume density and view dependent emitted RGB radiance at that location.
We can then use techniques of volume rendering to composite these values along a camera array.
So that's the cool part. Composited. This rendering is fully differentiable, so we're able to optimize this.
This notion of differentiable rendering is actually very cool. That's classic machine learning okay.
A collection of standard. Wow. So now you basically capture the whole 3D model forever.
Now you can make new camera angles on synthetic synthetic scenes.
Scene representation networks implicitly represent a scene using a fully connect.
Because there was no turntable animation, it was just simply a bunch of still photographs.
But you see, the artifacts, right? There are definitely artifacts. It's not magic.
So I show you a different one that converts all these two point clouds like a light.
Our self-driving car camera point cloud. So magically, the rendering quality goes up like crazy.
So this is not the state of the art, okay? In 2019, you can see or actually this is 2020 during the pandemic with limited samples.
However, okay. In other words, this older older technology this one is Nerf here on the right hand side.
Yes, Nerf is better than the older stuff, but Nerf also has problems.
We see the same trend in all of our synthetic. I'm going to move on our models with complex occlusion it secularity when it first came out.
This is actually highly surprising geometry. But I go back away.
So at North Carolina there was somebody that made a graph paper from many years ago, 1990s called view Morphing.
So view morphing is the very first time CMU, I got that.
Okay, maybe the guy moved on and I know it's not from Samuel, but look though, so cool.
Right? Okay. Wisconsin-Madison that other words one input photograph second input photograph.
And this was entirely synthetically rendered 1990s.
That is a start. Not even nerf. Never seen at the start.
I was at Dreamworks when all this happened. So we actually tried to implement some of these camera models.
Okay, so this all fully works by the way. You know, it's all computer vision stuff, but it's so cool just from like one photograph,
the guy looking up and the guy looking down, you can actually generate some intermediate views.
You know, it's very fascinating. Okay. The Mona Lisa, you know.
All right. Okay. So I'm going to keep going.
Yes. So Nerf explained. And I've explained Nerf to you. What else?
Um, yep. Okay. A little bit more. I should show you a 3D Gaussian splatting pretty soon.
Yes. So in the end, the scene that article once again goes in the exact same immersive view.
Okay. Summary. Bottom line Tldr search will one day feature these so easily from your phone.
Great. Coming to a phone near you. What about this?
Microsoft. Video generator.
This one. Oh, Vassar.
Vassar. So you're an independent business, like a photographer or a marketing consultant or designer?
Oh. Well, just when we thought we knew everything they want to know about was us.
Extremely shocking because a single photograph, a selfie, some internet search of somebody, one photograph and then some audio.
You can make that person sing that audio, say that, you know, whatever the thing is saying.
So I want to play a little bit more talking.
To that. Just one. One picture is all it takes. Generate AI generated images.
We want to actually, uh. Okay.
As cool as that is, I'll actually go to all.
Yeah, yeah. This one. Actually, I think no one.
No. Not done. I wanted the main announcement, you know, that they made that pretty set.

Like Microsoft created scary new I less Azure.
Uh, what would that be? Maybe this one. Yeah, five days ago.
So I'm looking for something very short to play for you. A bunch of examples, you know, is not this.
So where do you think we can find it? Free.
Maybe at the top again. But you want the Microsoft link. Okay, I'll just say var and see what comes up.
The one denounced by someone in a blog post. I'm trying to find it for you.
TechRadar PC games, videos Microsoft. Yeah, this one, I think I found it.
Cool. Yeah, yeah. So check this out. Oh my goodness.
Okay so I'm going to hit play okay. So you know sometimes nothing happens and sometimes everything happens all at once.
And you just kind of deal with it. And it's also just strange to.
You can do facial expressions, eye movement things, everything anxiety, tongue, all of it to the highest they've ever been.
You know how you can make deep fakes so easily with this? China had an app called ciao a few years ago and then the government banned it.
Nobody can have it on their phones. You could do a selfie and insert yourself in any basically drama or telenovela, right?
Basically soap opera. But then that's all banned. But now look how easy it is.
And Microsoft puts it out there and, you know, somewhere deep in your skin window.
Murder. The first thing we need to look at is the letter H.
So that about five years ago generate deep, deep fakes were so difficult to make.
It took days and days and days and some random GPUs, and the image quality was so bad you had to basically take a bunch of images and crop them.
So only the face shows was a pain in the butt, so only the most dedicated people would do it.
But now just one photograph found one photograph. So just it means a scripted kid.
Anybody in the world can do it. Help! Okay, so. Okay, so all of it will be okay.
The my sellers, the milks, the cleansing bombs, the oils.
They are really. If you plan to go for a run and you don't have enough time to do a full run, do part of a run.
He never said any of this. You plan to go? Surprises me still.
I never said any of this stuff last night. Uh, it was fascinating.
It even knows expressions like fascinating. You know, it knows what the words mean, right?
And then it animates the face for you. You can read the archive paper.
By the way. They wrote a paper about it. I'm like, what?
He even does two voices, like a comedian voice and regular voice. Like, how do you convey love to your partners and loved ones?
How do you do it? Um, but you can imagine I have a lot of questions, so, um.
I'd love to begin with you. I'm not sure if she's British or Australian.
First, it is because I read that you Star Wars catapults will not only make your users journey more pleasant, interrupted, and they hate getting broken experiences. Okay, and there's more.
A little bit more. I would say that we as readers are not meant to look at him in any other way, but we can parameterize this.
That means you can actually control to a certain extent, you know, the video gaze direction, the camera orientation and all that.
Okay, so the parametric thing is, in other words, not just always this kind of talking head can make it.
So, um, I'd love to begin with you. Firstly, just because.
Do you stand up all the time? I'm a stand up. Go head up, head down.
Oh, no, I'm a paparazzi.
I don't play no Yahtzee. I go pop, pop pop pop pop pop.
My cameras pop your friends. Not to let you not think I'm sick.
You to sleep. Me?
Come on. Eat. Healthy.
Talk to me. Who would have some pride?
That's what has come to. Show me that you love me.
Oh, what a mess. Hey, Swifty. She got pissed off that, uh, some Google drive with her songs got leaked, right?

Wait till you get a hold of this. We can make Taylor Swift sing anything you want.
Not anything she wants in her voice, not yours.
Okay. I'm not surprised. Not everything laughs.
I've broken my heart so many times. But I am going to suck because I'm an individual.
What's your name, partner? Poonam Bano. Ugly.
And you don't look. If I were Chairman XI or Putin or, you know, Kim Jong UN, anybody?
I'd be pretty worried, okay. About what people can make me say. Make me say.
But he's really something that he's like, okay. Practice makes a person like that is the parameterization.
So you can actually set sliders and hit play and just watch that, okay.
Wow. We can even pick different faces to pick different characters.
What I decided to do. I decided to focus all my attention, all my time on listening.
So instead of doing something else, I just listened and listened and listened.
Because one day there can be a zoom call with you in the zoom call, except they are playing a video game somewhere, right?
And in your, your eye is holding on an entire call as if it is you.
Those days are not far away, by the way. Okay. How do you prove okay.
I'm a true believer that if you're really bad at something like.
See, that is the text. It's crazy that that's being pronounced by her text right there.
All right, you guys? Yeah, right. You know, I always love it when they're playing nice.
Okay. Oh, my God, this is all pretty amazing. Please read carefully. Okay.
It's actually pretty responsible at some fundamental level for them to simply release all this in the wild.
They know people are going to misuse it. Okay, but the cover the ass by some dumb statement like that.
Oh, please, please use it carefully. Same thing with OpenAI.
By the way, you know, incredibly responsible. Same thing with stable diffusion. All of them.
Really? Okay. On the one hand, they're pretty amazing. On the other hand, they're also weird.
So we should do our part three today. Before that, since we have so much time, we can have a lot of fun.
Let's show you want to. Oh.
And then we will do our tenants after this. Hey, what are you going to do?
What are you going to do? It's going to walk in. You're going to have the microphone.
Can't. This one. Meanwhile.
Sorry. That this is work.
Sorry. I was actually playing Frisbee with some people outside the.
Okay, so I'm thinking, um, you know, how we're implementing AI into all of these systems, right?
And, uh, we're building it all up. And then at some point, we're going to have to burn it all down, right?
So in honor of that, I'm going to sing Burn It Down.
Hi. We're visible after the AI companies with nothing to hide.
Seriously? Ones you don't have? Yes.
Not our thing. Great. What do you mean, good thinking?
For the fake mike just because. Oh.
You sucker repeated.
Uh, the explosions broke in the sky, and all that I needed was the one thing I couldn't find in you.
Where the rabbit turned, waiting to land.
Uh, we're building it, uh, to track it back down.
We're building it. Uh, burn it down.
You can. Way to burn it to the ground.
The colors can play. Today as the flames climbed into the clouds.
I wanted to fix this.
I couldn't stop from staring it down.
And you would have had to cut and burn.
Uh oh. And, uh, was there.
I had to wait until, uh.
No, uh, we're building it.
Uh, break it back down.

Uh, we're still doing it. Uh, burn it down.
Can we burn it? Do you mean.
Yes. You call me half an hour? Believe when you told that lie, uh, played soldier, you play king and struck me down when I kiss that ring.
You got that right to hold that crown. You up.
But you let me down. So when you fall, I'll take my turn.
Fanned the flames. It's your blaze, it's burns. Uh uh.
Rapid turn. Where you needing to, uh.
You know, uh. We're building it.
Uh, break it back down.
We're building it up.
Burn it down! We can. Way to burn it for you.
So when you fall, I take my turn. And fanned the flames.
Since your blazes burn suns. We can't wait to burn it to the, uh.
We can. Way to burn it to the ground.
Okay. Thank you. It was great.
Thank you. We have way too much fun in this class.
Okay. Uh uh uh, there's still much, much more to.
Okay, so we, after this fun little break in away is 633, right?
So, as promised, all of you came to class after all. And that's going to count for something.
So therefore, here goes. Uh, Bishnoi.
Also, people in the remote, uh, room, there's a whole bunch of you out there,
and you can please go and chat and say you're here, or maybe take a screenshot of Central America because Bishnoi here or not.
Okay. Hopefully in the other section. Okay.
Like I told you, I know that Dash five of the final is actually tomorrow, so a bunch of people out studying.
But you know, you shouldn't rob Peter to pay Paul okay or something, right?
In other words, don't skip class at t k and all together now can all k.
Nobody cares. Nobody can. Okay. Oh, no. At least as far as I can tell, there's two out of two.
Wrong. A recall is right now. 0 to 3.
Critical. Look at that. This is zero out of three.
Oh my God. Link. Yeah.
You broke the chain. You broke the chain, I like it. The cold streak has been broken.
Now where's the hot streak? Okay, I like it. Deng Link.
Uh, z1 z1. Uh oh.
This thing happened to be that odd person. An outlier. Okay, well, let's do one more.
I'll do. I'll do it. I'll keep doing it again. Okay. Lot of car, or is it all over the car?
I'm not sure. Thank you. I see you. Super cool.
Hey, this is good. I'm going to go on here for a second.
I'll show you something super useful. But this is not attendance, actually.
So I'm going to show you. Should you came in, I think, one time.
Right. K-means clustering. But now you know what we can do. We can do WebAssembly.
You know, I told you so many times. Okay. But WebAssembly, where this is all going even right now, you got your homework, right?
You got a Python notebook. But future applications will look like this.
The application can be, first of all deployed on any platform.
Can be an app can be a web page can be a desktop. Okay, so the place where it runs doesn't matter what it does.
To do all of this, you ask a question from a video file. It shouldn't matter where it's running because they're all running on different cloud GPUs.
But why multiple cloud GPUs? Because each is a separate function call.
They're all being orchestrated by this app. So the app sends data to this call and then gets a value back and sends data to this call,
gets a value back and goes like that and sends the data back to bottom.

So it doesn't matter what goes where, right? That's the best part about all this.
So one of the magical ways in which we'll get there, send them by the way from Nvidia,
it's actually that same name stands for Nvidia inference microservices.
So where again this what Jensen Huang announced where you can do actually a lot of applications as these kinds of a little package function calls.
This goes right back to history in 1979 or something.
The good all amazing beautiful C programing language. Not C plus plus not Java, not Python, not Ruby, not gore, not rust C even lower level than C.
C is beautiful because he has no object orientation.
That means you cannot build this pretty heavy class layer to which you become a slave once you build it.
Once somebody builds a massive C plus plus hierarchy, right?
Some new developer cannot come and change any of it because there are millions of lines of code that uses a hierarchy.
Okay, people don't realize that object orientation is great, but slowly it's losing its shine.
Instead, in the new world, what is more powerful is like a gazelle.
A gazelle is a pretty lightweight animal that runs really, really fast.
Okay, the reason why the gazelle runs so fast is because it is not traveling in a big pack with all the animals.
It couldn't care what happens to your friend, okay? It just runs as fast as it can.
Other words, individual lose little pieces of functions.
And because these functions run independently on different servers, there's no state unless you pass the state into the function.
Those programs are servers are not talking to each other. Okay. Or maybe app maintained state.
In other words, be as low as possible. So the world's best way to do that is use WebAssembly.
So with WebAssembly you can write a good old C program,
a very simple Hello world printf C program and convert that to a WebAssembly file, which I'm going to show you called wasm file.
The washing file is one of those. This is a wasm file. And the washing file has one function call.
And then if you call the function with a function arguments, it will return you a value.
That's all there is to it. Wasm is pretty well documented.
It's a standard C that's fixed data types.
You'll know exactly how many bits the integers are, how many bits floating point is how many you know for a string.
Like what? Whatever, right? It reminds you so much of the Java programing language.
Java sadly died. Java was screwed up by Sun Microsystems first and then by Oracle afterwards.
Java still survives. It's very sad. I get the Java magazine right.
It's basically and also ran, you know, basically just please just go away. Okay.
So someday it will go away because these kinds of things will take its place okay.
So lose functions. Even better, you can program the Wasm function yourself in plain text.
You don't even need C programs. You can use C rust, Go dark C plus plus Python.
It's a whole bunch of languages are all Wasm compatible. You write the same function in any of those languages.
They will all result in 100% the exact same Wasm binary.
That's the beauty of it. It means you can mix and match these calls.
So some open CV call can be in Python, some other call can be in Julia.
For Turing machine learning inferencing. You can combine them. Just pass the data get results back okay.
So to show that this all fully works, now I'm going to show you why some underscore and check that out okay.
So WebAssembly ad all I'm doing is downloading my CSS file which is basically nothing.
And then print file which is nothing but this one. Fetch.
Check this out. Such bytes dot USC wasm add that wasm that add that wasm is what I just now fetched.
Okay, from here that contains. Okay, so I have to fetch it.
Then the response is array buffered. And so this is simply boilerplate.
All of this is 100% boilerplate. All of this in that wasm file was an add function that I wrote.
And it's adding two numbers. Watch this okay. This. O-m-g!
Cool. I'm calling the washer Matt from JavaScript.
Okay. I cannot tell you how crazy this is.

That could be a self-driving car. Data live traffic within the parentheses.
And the result is the car driving itself. Please, please think of that.
So then what does the washroom file look like? It's binary, but I can show you what the.
What file? If you program directly. Not using any language directly in the washroom programing language that is called a word file.
WMT file was some text file. That is beautiful.
Looks like assembler can write internal functions. You can write that function with it.
And then then no need for C okay. No need for compiling. How cool is that.
So watch this okay. Truly play with it.
I mean, really. Oops. Wow.
What was that? Okay, so I'm going to grab this. Actually, just go there.
That's what the Wasm binary looks like, right? Okay. But that's because it's been compiled from a C program.
But what does wasm ad look like. Wasm.
What add function. Add example of something. I want to show you the world's simplest all atom.
Oh, I found it. Get ready. Oh, joy.
Think of. This is a brand new programing language which we should all learn as a programing language.
The language helps you declare like an entire function, but the functions name, functions, input, functions output.
That is all a function ever needs. What is a function's name?
Dollar ad. What should the consumer call it? AD.
You can rename that if you want it. What is the first parameter to iterate an integer?
What is the second parameter like? I had a comma b. What is the result? Well that's the whole function signature you guys.
That is all it is. And what is the function actually do? It loads.
It loads one of them back into add. It gets uh actually sorry.
It calls the built in ad function on dollar and dollar B.
All well, there's no return one or whatever it is computed.
So stack based by the way you put it on the stack and this tax output gets returned.
This is pure magic. You can do computer vision with it I'm not kidding.
You can do machine learning with it. So gradually PyTorch Julia you know all of that will just go with TensorFlow.
Just go away in the native language in Python Julia did all the cumbersome calls.
Why? Because then we can do this. We can cross platform mix and match them.
Say I have once again say, after this I have some other call.
In other words, here I am doing pretty right. I don't need to pretty.
I can literally receive this somewhere. I can say let x equal to you know that's what I'm saying to you.
So then let x equal to. Except I do not know exactly what I did.
Meaning I don't know how many parentheses. Let me see. Um then obj or.
Sorry. It just kind of went off intention. Okay.
I'm going to try and keep it very short. But this print statement. Right.
Let's remove the part. Okay. I guess I'm trying to show you how you can receive this in a variable.
Okay. So now I'm calling it and throwing the result away which is fine. I'm going to say let result equal or something or is equal to.
So now the whole idea is I can print yours. In other words you call the function.
You got a response back. Cool. Now the results back.
So this is one function call that I'm making. Say I wrote this function in Python.
After this, I can send this result into a new function that might have written in the source language in the rest mixed language programing.
It means you can use any language for its advantage. You can use Python for an open CV or something, or you can use Julia for machine learning.
Great, right? Got a question? Could you go for a walk while?
That's a great, great, great question. I honestly don't know, but it must be possible.
If you write it, people would love you because then they can take. Then you can do a cross compilation, right?
Take rest. Make it go into what? And then from there, go into Python.
You know, it would be a great idea. Also, can you reverse engineer some files and turn them into what files?

I think that's possible. Yeah. You can do crazy things, you know.
Okay. So again, you know, I don't want to go off on a crazy tangent here.
Right? But look at this. You can actually do like crazy, so. Well filter.
Right. So Sobel filter is an actual image processing filter.
I'm not sure if all that is going to run here, but at some point you can actually run Sobel.
Hmm. It means it'll grab your video camera and actually in live, real time, turn that into a filter.
Just a demo. I don't know. Try this and maybe come back.
Okay. I'm going to shower because I want to show you the new stuff.
Um. Ah ah, yes. You're going to see this.
Ah ah. Jill. Yes, yes. Sobel.
Demo. Yeah, there's a lot in here.
I don't know where this all came. Okay.
This is what you're supposed to say. Meaning you get to see, right? Imagine writing this in C, and this will entirely get compiled to Sobel dot asm.
And then you hook this up with a little bit of JavaScript that will take video frames from you,
up from your webcam, and call that Sobel filter that you see right here.
You can do serious image processing. See Sobel.
Okay, call like Isabel calls me. So it means time has come.
This OpenCV, which you can take any code that you already have in any programing language and turn them all into awesome.
So Nvidia's name I told you, I'm not saying all of name already in Wasm, but it could be.
So you can then run all these crazy models. Oh I forgot.
Wait, we're not going to the slides. So in the slides I'll tell you about small language models.
Microsoft released something called slide three. So pretty soon your generated calls will not look like a Python notebook is the point.
They will all look like Wasm calls, what they call as documentation name of the function inputs output.
What more do you need? Nothing. You know. Yeah. Amazing, right?
So with all of that out of the way, I think we should finally do something that is,
uh, from our class, which is what is my last set of slides I have to show you.
Okay. Great. But it's all relevant one way or the other.
You know, so I'm not entirely going off on crazy tangential useful. That's what I would do personally.
All right. So you know part three, right. I've got about eight slides and we can take it easy.
Knock over them all. Okay. Cool.
So the first thing that I have to show you is you need free up some space there.
Cool. This notion of attention.
And I play some videos for you, you know.
So the biggest thing that revolutionized so-called I really compared to what used to be imagined as a watershed moment.
Yeah. Time before that.
Recurrent neural networks and maybe their cousin called long short term memory Lstm.
That was a state of the art for what is called sequence. To sequence.
Um, transforming.
That means the neural network is given maybe some words with three words for words, and then its job is to turn them into 3 or 4 more words.
Tokens become tokens, tokens and tokens. So we call the token transformation like word, you know, not word to act.
Okay, so I'll show you I'll show you how to read and works. Suppose I was a state of the art.
Suddenly, after this idea called attention computation was introduced.
And then it is variation called self-attention, which the transformer paper introduced.
Transformer introduces thing called self-attention. But attention came before self-attention.
Attention is from 2015. Actually, believe it or not, okay, that radically changed how this whole sequence to sequence, you know, transformation happened because it uses a slightly new form of what are in a new studio.
Okay. And then when that became self-attention, this paper called Transformers came out.
This is 2017, 2017. That was completely a watershed moment.

And then Google actually followed that next to it, something called Bert really,
which is an adaptation of transformer, and then started using that in all that Google search engines, you know.
So that is being deployed as we speak, Bert, today we are in like way, way more powerful things and way more search going off on so many tangents.
That's why I have eight slides to show you. Okay. So then we're going to go off this way okay.
So I'll start with this. So what is this whole notion of sequence to sequence in know transforming even mean.
And then talk about attention a little bit. So let's spend some time on this.
First of all this made by, you know Julie Wang. You can actually follow her on like LinkedIn.
And then she actually is showing you with this amazing diagram. Transformer is not even just one thing initially.
Yes. Transformers. The people that Google wrote, right. It has variations like encoder decoder decoder only, encoder only.
But you know. So Bert is the whole encoder only.
Right. And then Bert has all these variations in some of your ML courses, maybe in like you one,
five and nine and kind of special competition you might come across think like Roberta.
If not, spend time and learn about each one of them slowly in your leisure.
Meaning during this break, if you find lightning that I.
Some of them might even have things like Roberta, a little model you can run or going to Huggingface and you can find a tutorial.
Okay, so it's good to learn the differences between not every single one of them, but some of them.
These two are highly useful. So then coming to this direction transformer decoder only Gpt3.
So in um, Google wrote transformer in 2017, OpenAI I started doing this part.
They secretly, quietly announced I was their fault. Remember?
They even made noises about how powerful this is in the world and they wouldn't release all of this code.
They were releasing little pieces. Okay. And that one when GPT 3.5 came out, that is when ChatGPT was made.
That is November 2022, literally. And here we are, just not even full.
Two years. Okay, suddenly the world is radically changed. But that is actually what is called a decoder only transformer.
So the idea is that there are so many transformer variations also.
So Uli has this pretty cool article which I'll click on called A Good Look at Transformers.
But I want to start by giving you this the basics. Okay, but I'm going to start with Hjalmar.
So Hjalmar again see this I think I'm going to play today.
J explains Transformers pretty well, including a half an hour video of which I'll play certain segments.
And then I have all of Uli Wang's article, which will click on, and then they also have Stefanie or the three people.
Right. Vinegar. Four of them.
If you spend more time than what I'm going to do here, just toss for leave the whole world alone, just us, where you will know 100% of all of this.
Cool.
And then Stephen Wolfram is for more fun, because here I can show you how when you splat all of that multidimensional encodings into two dimensions.
What are these words look like? What is textual encoding look like in 2D?
We can look at that okay. But don't click on it yet. And then this one is very nice also because the math that is explained in 123, four,
this first part is shown in a slightly bit more visual way in which, uh, in this YouTube video you can do that.
And this one is incredibly great, but it costs 20 bucks, which is absolutely nothing.
But I'm not going to play it here though, so if you want to spend 20 bucks, do the course.
He's one of the best people on the web. The guy had a half million dollars per year machine learning job at meta, which he quit and then made his channel and then starts, you know, making all these tutorials.
Okay, so he clearly knows what he's talking about. Highly beautiful, colorful diagrams.
So I'm gonna recommend that. And, uh, yeah. So then we'll look at a little bit more of the, the hype all of this.
Right. And then other domains. So I'm going to start at the top. So Hjalmar I won't even tell you about Transformers.
I'll tell you about RNNs pretty quickly. There's a very short video that we're going to watch right now.
All right. Attention! Attention.
Yeah. Because you can actually see what this notion of attention means.
Okay. Yeah. CC sequence to sequence model is simply some words going in.

Some words come out in this context. You ask a query, then the result comes out okay.

But in general we call them tokens. Token sequences. Tokens can be pieces of words that can be code, that can be image pixels.

They can be lots of things okay. So it's a token transformer.

And then when you do token token based machine length human language translation, you give it a French sentence.

That's a three tokens gesture. Yeah. And that becomes I am a student in French.

There's no gesturing at Odeon in French business, I am student.

There's no arm. But in English, that's the AI. Okay.

So what is not happening is word by word translation between any two human languages, Chinese and English.

We cannot just take one English word and make one Chinese word translation.

It just doesn't work like that. In German. You pile all the verb said away and you got to pay attention to what the person saying.

All the words have got to end okay. So that is not what is happening.

It is actually able to know for every single word that is being translated, including sure, for sure, including sweet interpretation.

What else should happen? There is a notion of paying attention.

So you have to basically, when you take every word, pay attention to all the previous words also.

But that's slowly coming. That's now what this is doing here is doing one at a time in a way.

But so encoding and decoding okay. So what is happening here is the middle of the semantic that you can take a an English sentence.

In this case an encoded English sentence encoding means a lot like in order to convert here right.

The encoding turns the English sentence or any language sentence into a bunch of abstract tokens that we call context.

The context can be thought of as language independent. Meaning no matter how I say I'm thirsty, I can say I am thirsty in so many languages.

I'm listening to his water, right? So that concept of I am thirsty. Imagine it being turned into numbers.

That is what those things in the middle are the context.

Once you have the context, you can reverse it and say, you know, generate the same context into a new language.

You can even go back to the same language and see if you got these words back.

Okay. Meaning language A encoder, context language B it's a pure language translation problem.

That is actually what the initial transformer was actually for. So a transformer because it transforms human languages.

But OpenAI realized, oh my God, you can actually train all of the English language with this and make it generate tokens.

That is when the whole GPT was born in GPT generative pre-trained generation.

But except OpenAI did not invent the whole generation idea I was administrating.

The next slide again. Okay, so we have a lot, but I'm going to keep going.

So again, the notion of encoding decoding, they're just simply again words to vectors and vectors back towards this a different neural networks okay.

So context is a vector ultimately I told you right a bunch of numbers. So one language will become a bunch of numbers.

But any language could have become similar numbers, almost the same numbers. And then you can reverse that and become a new language.

That is how you do language translation. Without this context, we cannot hope to ever go from Japanese to English, Russian to English in the US.

The government was obsessed for decades after that, during the Cold War, after the Second World War,

to take Russian scientific nuclear journals or to publish a few things that they want to publish.

And then how do American scientists know what the [INAUDIBLE] the Russians are doing?

Nuclear power. Okay, but there's no way that Russian English worked it out.

Tried and tried and tried and failed. It all failed. It failed because we take a regression sentence and try and do what's called a parse tree.

Noun, verb. Adverb. Yeah, good luck with that. And they are trying trying to parse tree into an English.

Think again a little like this. But it's not these words actual grammar, parse tree.

And that never ever, ever worked. There was just toy only work. Okay, so no actual scientific documents.

Same with Japanese. Any, you know, strong language that is not English.

They published, but there's no way in [INAUDIBLE] of English. This is slowly changing all that.

Okay, so RNNs and analysis are not that answer very similar, right?
Just wait. Each word can become its own vector actually.
Then you combine all of them and you make like the target word. Oh here's RNNs okay.
So like here I'm going to start from scratch. Okay.
So once again, in our minds we use these extra data that we get from the input called a hidden state.
So state is not new data. It's still coming from the input okay. So the input along with the states can be used to transform to an output column.
You see there's a little loop going on okay. That's what that's called recurrent.
So recurrent unit. But it can be unrolled to make it be like multiple stages of like a neural network.
Okay. So you probably know all this, but if not we'll take a look here okay. Time step time step time step.
This one is not unrolled. Yeah. In other words, just way through the and is becoming.
I am a student. It has the extra aider by the way. All right.
So when you unroll that loop that you saw previously, it is simply stages.
First stage. Second stage. Third stage. Liquid stages. In transformer today we're stages.
We're very similar. We call them stacks. Stacked encoder decoder okay.
Okay. So some things are still similar, but one. One. Ha.
This notion of attention okay.
So then if you ask what is so magical about attention what it is, is in here you basically only one token by token for each input token.
You worried about what output token is.
But by the time you got to the last input token and you want to convert to the last output token, you are not paying attention to the previous tokens.
You forgot you moved on. Okay. So the big thing that attention is doing is you want for the very last token that you want to answer for,
you can always go back and then examine the previous tokens.
That is all attention is okay. You're paying attention to all the like when I am talking.
It is not like you only pay attention to the last word that I said.
If you if you hear me like that, you wouldn't understand a word of what I'm saying, right? You keep your story in your head.
What? I told you at least the last 10s that is actually what is happening.
So we call that attention computation, okay? That is the only difference, believe it or not.
Okay, so look at this. Okay. Pay attention. All right.
A time. Step in time. Attention. Oh.
What's that? Okay. This one. Some model with attention basically meant to say that's where the new stuff started to happen.
And transformation or simply extensions of these. Okay. See this.
So you're not just getting one hidden state, you're getting previous hidden states also.
Cool. Wonderful.
So somehow that one difference became like all, you know, all the magic.
So again, I'm going to not start this one. Okay.
I am leaving out lots and lots and lots of math. But you know, that's okay. I'll show you more where the math actually is.
There. We're just getting started. Really? Okay, so, you know, this is actually very cool.
This notion of a softmax softmax will turn all of these outputs that have not become words yet into probability.
And then you look at the highest probability. Each is going to become a word. This could be a word.
This could be a word. This could be a word. You take the highest probability.
I look up a word. Each number like in a dictionary. The number becomes a word, and your word call to the highest word.
That's when you when you spin the wheel and say, generate me one more in output.
It will then maybe use the second highest probability and then return whatever word it could a journey.
Okay. That is why it makes variations one word at a time. But the words are all probabilities, okay.
It's a distribution. Okay, cool. I haven't shown the actual transformer paper.
We're going to go there. So again sequence second model with attention is all about attention.
This one is 30s I'm going to play it. And then we should do the actual transform.
But then take a break, okay. Promise you.

I show you the attention computation diagram in the transformer.

Cool. Your input will actually become your input and become embedded and then become three vectors called the key vector, uh, query vector.

And they will get turned into a new vector called a value vector.

And the value vector is the one on which you're going to compute the softmax and output the token.

The next word, except the next word will also become part of your input.

Now the input has become longer by one word, and then that will get embedded and become a quick again query.

And then pull out the word following that first word. Now you have two words, and then the second word also goes back.

Input is getting longer and longer and longer, you know. And then put can only be up to about 32,000 tokens or 64,000 tokens.

So that is how the and the words make more and more sense when generated.

Because initially it only generated one word, but the word generated helps you generate the next word that is called autoregression.

That is autoregressive. And so then the input is getting longer and longer. Cool.

Okay. So all that is actually coming somewhere. So finally you know in here right okay.

Look at the attention. So it's all about attention. So especially when it gets to student add a student okay.

It is not doing Germans I assume means um student.

Then you will get bad English call I am student. You need I am a student.

So watch what happens when it gets a student, okay? Just pretty simple.

Just becomes I. And then am always am.

But now look it added that a and then student.

That's pretty cool because I hidden states helped it. Likewise a more complicated sentence lower cause to lessen economic, European or destiny.

So then, you know it says economic European. So that is French.

In English it is European economic. So the word order can also be swapped when you do the translation.

So it is not doing word by word because then you get bad English okay. Okay.

So you can obviously do a lot. So I'm going to get to the Illustrated Transformer now and show you a glamorous amazing deal.

This one. Over the last two years we're going to spend a few minutes on break time.

Community have been building bigger and bigger models that are doing more and more impressive things.

Not enough for you to actually get what is happening, but I have more process where we've seen really impressive models like the GPT three.

You can refer back to my previous video to learn a little bit more about GPT three.

We saw a couple of the demos, the examples that the model is able to do with the website, uh,

or learn how to build websites, I guess based on a few examples, is a type of machine learning model.

It's a so that is the amazing paper that they wrote, right? Two of them are from USC.

Ashish was one and Nikki Palmer, they both used to be at ISI.

They actually better be researchers okay. And so then they went to Google, wrote this paper.

Well. And, uh, so I'll move a little fast.

A transformer basically is a type of machine learning model.

It's a an architecture of neural networks.

Since then, variants of the transformer, namely a variant called Bert that builds on top of, say, truly took off.

You know, people made Bert, Roberta, you know, Estelle. Bert models on and on natural language processing in various tasks.

These are some leaderboards, so I'll get to the actual part where it talks about Transformers.

Okay. Vision Transformers that are they can do image stuff.

Also iteration transformer transformer I think ten years ago.

So then that paper, that, uh, article right there is highly worth reading that blog post.

But, you know, it's going to be he's can explain that a million page views.

Uh, I believe that same same example since that that English sentence just some of the visualization.

So I'm going to keep moving blue box in this case that takes a sequence.

Uh sequence three. Uh but we try to predict your query.

And the amazing answer I give you puts it or GitHub copilot code comment actual code with IMA student events.

Given that that that French sentence. And that's the example from the initial paper because the initial paper talked about,
uh, machine translation, language translation, human language cognition,
that, uh, look at that general black box into two smaller components,
two smaller black boxes, let's say, inside of it, you know, one is an encoder stack.
So the input goes to the encoder stack and then uh, output some of its processing, uh, results to a decoder stack.
And the decoder stack outputs again one language and then context in the middle and then output.
And then we mentioned that the stacks each of them is composed. So you can then stack them okay.
And then likewise you can stack them as well. In fact you can also when you have multiple GPU cores, even compute this word that we call attention.
These these numbers call attention in parallel, you know, so that we can go fast.
Actually, you know, that is called multi-head attention.
Head means computations are multi-head means in many computations happening at once, you know, multi-head attention.
That's simply the speeded up of layers, uh, six layers each in.
I'm going to keep going. And the gpt2. Yeah, but the GPU to do is one variation called the decoder only model.
There's all kinds of different variations that you talked about in our diagram.
Right. The only variant. But then let's actually get to the architecture, which I believe starts here somewhere.
Learn about the encoder or the encoder decoder model. Now it's going to happen.
So our focus here is going to be the one in the middle. The cool code.
Look at that one. And then these models are put in in.
Actually there's not even the attention computation I'm going to get attention computation.
Uh essentially what happened. So OpenAI took the decoder only transformer and then made GPT three by doing 300 billion tokens.
Just imagine how big that was. Now it's up to 1.5 trillion, 2 trillion tokens.
And Chinese models are even bigger than US models. So token is roughly a word that can be a piece of a word, even.
Okay, part of a word is a token. And then its goal is ultimately for all the power, the magic that lamps have on our lives, it is simply outputting one more word, not a whole sentence, just one more word.
And then that word gets added to the input. And then the goal is the same again.
One more word, which is my second output, and that goes in the input.
And then one more word.
You know, among other things, it has no way to go back to all the words generated and said, sorry, my sentence looks pretty bad.
Or, you know, I hallucinated it is outputting strictly in a forward direction one way at a time.
It is pretty shocking that it works that well actually, considering it's only one writer so he doesn't actually understand what you're querying.
First of all, it also doesn't understand what it actually produces one word at a time.
But to us humans who can make sense of all of this, it looks great.
I'm going to go. Okay, this is a great question. Say like predict the next word.
A robot must a robot must kill everybody.
Um, a robot must work. A robot must go to sleep.
You know, a robot must, uh, I don't know, eat. There are so many completions possible, right?
But because of Isaac Asimov and all kinds of robot science fiction, most likely a robot must obey the three lots of robotics.
So maybe the next word is obey. But the other things I told you are also possible.
A robot must, uh, work hard. A robot must work flawlessly.
So there's words like work all that, right? They're still part of the output.
So they have a probability. The best word possible is the robot must obey.
But if you say, can you give me the next higher probability? It might tell you a robot must never fail or something.
So let's see. Okay, that is all it is simply going to the past data and then searching through, so to speak, what the next word should be.
It's a pretty sophisticated search, but it's not idea of search.
It is a search in this multidimensional embedded context space.

That's actually all it is. Okay. It's still search okay for the nearest neighbor search similarity search.
We just got a lot of text from the internet, from Wikipedia, from various websites.
Okay robot, what is the most training? Samples from?
How do these training examples look? They look like this. Ha!
And so let's say we have that text at the mask and coding mask.
A mask training. All right. So in classic machine learning we take a table of data.
Some table of data. Let's say maybe students.
Courses. Students. GPAs. Students. How many units in our undergrad GPA?
And finally we will label the student as successful or fail.
So we label this last column that is our, uh, output.
It's called output column.
And we train the whole neural network to say if you encounter a student with this kind of profile, predict that they'll fail USC.
And why do you say that? Because all the students like them, that's what they did, right?
So we need to create the last column that is called a labeled column.
So this makes it called supervised learning because we have to create the last column here.
There is no last column is unsupervised learning. But still it has to learn right.
So how do you make it learn? You basically hide the last word and make that to be labeled data.
In other words, I know that the next word is a. You tell me.
Okay. I know that the next word is the second law of robotics.
A robot must. So I know that by heart. So. But I'll hide that from you and make you guess your the neural network that is classical machine learning.
So if you tell me, um, a second law of robotics, a tomato, that's a big error, right?
Then I'll back propagate and then make it go on training over and over.
Predict, predict until it learns to say, a robot. Ha!
That is how we train. So we can train using unsupervised data by masking out words that we know that is like labeling, where we are not label anything and making it guess makes sense. Okay.
The top cop before we got the homicide DPD.
That's what opening. I had to do it. We can slide a window on that and generate examples, because we want to train the model to predict the next word.
So the first let's say example, we say is that okay we'll use the first for this.
What we wanted to say, that word we wanted to say and hold these.
And after that we wanted to say robot. We mask them and make it guess uh, and so that's one training an insane amount of training.
Right? That's why it needs like $15 million worth of GPU power and electricity, because it's not easier, longer sequences of input.
Okay. But it's very cool. The training is pretty obvious. In other words, we have like no one paragraph at that present.
So we take non paragraphs and train that. So in this case GPT is already trained.
So no wonder it says a robot must obey. In other words it's almost like it memorized it okay.
But it is not searching through like actual vocabulary with a PDF file.
It is in some embedding. It's in some kind of multidimensional space. A very sophisticated search if you will.
But it is search. That's why I have all the words that it could have said.
It said obey because it was punished previously, which we didn't know but meaning were not there.
It was trained them uh, in a fashion like this. So if we have this example, cool robots will take a break soon.
But I want to think the words after the selected.
Uh, okay. Suppose this whole training works. So supposing this is a training step.
Okay. So you want it. That's that's what you expect. That's the label the last column there.
But then it's going it's giving you something incorrect. So punish backpropagate.
Error. Back propagation. Deletion. And it outputs a word and it will be junk in the stack because it's based on.
Um, you know, it's randomly initialized. I was going to be the neural network before it's trained.
All the weights are completely random. So to support any word at all. But, you know, you said troll that should have been obeyed.
That's the error signal calculated the error, the difference back propagate.

Change all the weights to uh, and try again. Put that into a numeric value.
After we calculate that error, we have a way of feeding it back to the model,
updating classic backpropagation that the next time it sees a robot must,
it's more likely to see obey, still not obey my might do this and punish it again.
Over and over and over. Millions, tens of millions of times. Tens of millions, all of the data that we have.
And then we have a trained model. Wow. And that's just crazy training.
So it knows nothing. I want you to know that, okay. Anything that it says came from words like this.
Let's talk about transformer language course. Why don't we take a little break okay.
So why don't we take a break for ten minutes till 721 and then come back and do all of this?
Okay? So please take a little break. We should.
We. Wish. You want to come and sing.
Yeah, I love that. Oh. Too many meetings.
Missing the key action items. It's time. Anyone can join from wherever you are.
Just come to the front and say okay, it's a good day to.
Have to. To you.
Doo doo doo doo. When I find myself in times of trouble.
Mother Mary comes to me. Speaking words of wisdom.
Let it be. My hours of darkness.
Standing right in front of me. Speaking words of wisdom.
Let it be. Let it be me.
Let it be and let it be. For it will be.
Let it be. Was there when the broken hearted people living will today.
There will be an answer. Let it be.
For me. There is still the same.
There will be an answer. Let it be, let it be.
Let it be. Let it be, yeah, let it be.
There will be an answer. Let it be.
Let it be. Let it be. Let it be.
Let it be. Words of wisdom.
Let it be. You.
Can you? He.
You. Let it be.
Let it be. Yeah. Let it be. Whisper words of wisdom.
Let it be. And when the night is shining.
There is still light that shines for me.
Shine until tomorrow. Let it be.
When you wake up to The Sound of Music.
Mother Mary comes to me speaking words of wisdom.
Let it be, let it be.
Let it be. Let it be. Let it be.
Well, that would be an answer. Let it be.
He let it be. Let it be, yeah, let it be.
Me and answer me. Let it be.
Uh, let it be. Love you.
Yay! Go fish! Guys, I know Devin's going to take a job as I'm in the midst of applying for residency right now.
Emily has again been a lifesaver in helping me write my personal statement and CV.
Grammarly is here and I cannot take this away.
Okay? Yeah. You know, live performance is actually going to be a thing, which was great.
By the way, little do you guys know, which is actually an undergrad student.
Taking this class for grad credit. Amazing. Okay.
Need need, need. So we are going to do more transformers.
Okay? Just without skipping a beat. Eric718 I can start, right?
Yeah. Major components of a transformer block.

I'll have. I want to play all of them half an hour. I want to tell you how the attention is computed.
Then we can move on. I would think of a transformer with only one.
It's extremely cold. It's a big deal. The BFD presented your attention.
And self-attention, what they call self-attention. Uh, this is so cool.
Now you know what's going to happen. Okay. The word Shawshank is an incredibly uncommon word.
And in Hollywood, there's a classic movie called The Shawshank Redemption, which you should all watch.
By the way, it's a classic Hollywood movie, so there is almost no word that is basically expected to match other than the word redemption.
And you trained I somebody already marked out the word redemption and punished it until it learned the word is redemption.
Just going to repeat that okay. That is all it does. The film The Shawshank Redemption and we what we said okay.
Predict that's the movie, right?
Morgan Freeman when he was much younger, the model has looked through, let's say Wikipedia has been trained on a lot of text.
And so IMDb, will it be able to predict what word comes after?
This is so cool. By the way, you can actually go right now to distill GPT two.
See, unlike, uh, GPT 3.5, which is credibility, and GPT a three Pro, which is 4.5, which cost money, Huggingface took the older GPT two,
which is still pretty powerful, and then made an even smaller model that is called distilling, by the way.
So in a distill and LM, you take something that has many billions of parameters and make it into a much smaller, lighter, slimmer model, which might be slightly inaccurate, but you can run that on your laptop.
Okay, that's what distilled Jupiter two is, so you can run it yourself. Cool.
Okay, we can actually try this such great.
Right now. Just go to their website and type The Shawshank and see what happens of the GPT.
What will it predict? Should we go there and we say, okay, literally type of Shawshank.
That's the our what is so cool is it didn't just complete the one word.
That word that it completed is correct. First of all, the word got added to the input.
So my new input is The Shawshank Redemption. And then predictably, it went into my IMDb description.
Right? And instead of pulling out all the words, okay,
this is why people actually sometimes think that it cheats in the sense it is doing some Google search.
There's no Google search. All those words next to each other. It's what it learned.
Okay? Based on like past data, meaning there's almost no other completion, but it actually makes it up one word at a time.
That's pretty amazing. Okay, praise. Uh, the model is able to generate a whole.
It made me interested, really, in the search. It's impressive.
Right? It was able to actually put in okay.
This actually what happened? Okay. So you know this attention is being computed, right.
That why is important. That word is also important. So the token the token which is basically the number representing the word redemption is here.
Right. That's what the decoder computes the token. The feedforward neural network is what turns the token, or a sequence of tokens into actual output.
So if you look at the transformer architecture, that's actually what they have. Always have a decoder feedforward decoder feedforward.
The feedforward job is to produce the actual output okay. Cool.
That's called a feedforward neural network. It's also called, um, the MLP, the multi-level perceptron in multilayer perceptron.
Multi-level perceptron? The previous one's own.
And we've had okay, so this is all extra extra income because, um, it's all the same.
Yoshua Bengio it's a more complex model. So your Bengio his name was in here, right?
He's the same guy we saw in the beginning part of a class. Okay, that he gave the interview at Mila.
Okay, so the next one is it has got the word it. And that used to throw away runs all the time.
It failed. So when we say the word it after saying a bunch of things, we have no problem following the whole chain of thought.

We never question, hey, wait a minute, what is it again? Right? That's what's going to happen.
So the chicken, the chicken didn't cross the road because it was too noisy.
Suppose the chicken didn't cross the road because it was too noisy.
There are two nouns. So the chicken and the road. What was too noisy?
Okay. When we say it was too noisy, then you can even ask the machine what was too noisy.
Hopefully the machine will say the road because roads can be noisy.
Because the roads are cars going right. You get that from past data. Somebody in a story road.
The road was so noisy I had to shut my ears and cross the road. So that is why the word it is decoded.
It's the word. It is ambiguous without attention, without paying attention to all the previous words.
Right? It can mean either the chicken or the word. And then which is more important in this context it's a louder.
So when you say it and the word loud, it can only apply to road.
So roads probability went up in the softmax and chicken's probability went down.
That is why it's correctly going to answer. You know uh just watch it.
It will tell you why I didn't cross the road. Okay. And so that was a pretty big deal to break that barrier that we had each of our disambiguation.
How hard is to know this idea is not specifically novel?
Oh, I'll just pause and ask you guys a fun question. Okay. We definitely in our heads do not do any of this.
We don't compute softmax. We don't compute attention. We do not compute. There is no computing zero computing in the brain.
So how do you think that we are able to disambiguate?
Like if I say to you, um, the chicken didn't cross the road because you has too loud and ask you, what does it mean?
Why would you tell me, ah, how are you able to tell me? Road. What goes on in your brains.
Just guess or answer me. Why?
Yeah. Yeah.
How? You're right. How? Yeah, that's the keyword, the magic word experience.
Okay. We live in the world, so we cross roads all the time. We hear roads being noisy is not data for us, okay?
It's actual experience. Really? Exactly. There are cute videos on YouTube about ducks and their families.
A little mommy duck and then baby ducks doll patiently wait in a big crosswalk.
When cars are whizzing by, they don't go. When the light turns green, they go.
It's very cute, but it doesn't mean they know what red and green means for them.
It's simply that the road stops whizzing. I know I can go, yeah. I fight you?
Seriously. You know, we don't want to turn our brain into a computer, okay?
I mean, the big mess up. Okay, you know what it stands for?
The big stuff that I did 50 years ago, and they never recovered from that big ass up is to do that.
So I'm not scalding your arms. Scalding all over. I'm scalding.
Scalding Yoshua Bengio scalding all of them is to make that deadly mistake that the brain is some kind of weird computer.
Okay, your brain is not a computer at all.
If we chop the brain open, you see blood vessels, you see chemicals, you see like molecules, we see electricity.
What we don't see is a clock with the cycle and some kind of bit pattern that's being decoded in blue and then added load store,
you know, like I'll show you at the bottom right. There's no load parentheses.
There's actually no explicit numbers. There's not decimal ten based based on system.
Again our brain does not do numbers at all. So we don't compete you know.
So that is why when you ask me to represent is an experience some kind of higher dimension.
Now the experience can be moral probably can be moral maybe.
And maybe it's right. We do computational neuroscience and on model like nerve, you know, uh, what I call what I call the call trains.
Pulse trains. Yeah. But the model is not territory.
In other words, just because your model works, it doesn't mean that's actually what the brain does.
I'm quite fond of saying that, uh, because you brought that up. I'll go on the one tangent very quickly and we'll cut back here, okay?
Because you might even wonder what does what the [INAUDIBLE] does even talk about?

How can something not compute? I'll tell you. I'll play a video for you if you take, uh, this one.
Okay. Take a digital timer. The digital timer can be incredibly accurate.
Down to 1/100 of a second. Your phone has all kinds of timers, right?
Okay. See that you can compute time to, like, 100th of a second.
So precise. Right. Okay. That look at this one.
Russian. Cool. Mhm.
I'm going to say hourglass and. Okay.
What the science is showing us is the universe appears to be alive.
Because then you will know what I'm talking about. What I mean by not computing.
Actually make the first year a chart with the graph it on the top.
Huh? I also. Okay. Suppose you don't do all the crap.
Okay. You can buy, by the way, so you can get a little shower time and on basic in the dorms.
Right. There are exactly a five minute timer. You set it. You know, when the sand goes down in five minutes.
Stop sharing. Save some water. Okay. When the timer.
When the sand is falling. You have to listen very carefully to what I say to you.
When the sand is falling, shook it in midair. My God!
Shake! Stop that hourglass for the whole. Oh my God.
Pretend all that is not happening, you guys. I just want to show it to you. Okay.
When we reconfigure an hourglass. Just leave it alone, okay? But then, of course, this is our, in fact, made from two molding that fit.
You know when to stop her householder. Okay. When that is happening.
Householder action when that is happening. That is how it counts.
Time. So a one hour timer. I tip it and all the sand is on top.
I watch the sand fall for one hour after one hour.
The sand would all be on the bottom and now one hour is up. That is some kind of computation, double cross computation, some kind of why?
Because it's marking time. It's telling you when? When I was over.
But there is no explicit computing.
There's no numbers. There's no variables, no data structures. You show me where the [INAUDIBLE] is the data?
Show it to me. There's no boys, okay? Don't dodge my question.
Don't ask me another question, but show me what the variable is. Show me where this a minus.
Minus should mean is nothing. It's simply material behaviors.
Phenomenon. Okay, that's all it is. Okay. But then I'm going to bug you with one more question, but not yet.
So this is what I mean by intelligence without computation.
You don't need to compute everything at all. You can still be intelligent in large time.
In the old days, people actually might take a whole day with this.
The kinks cord would end when eight hours go by, when all the same goes to get up and go home.
There was no pendulum. There's non-digital time. Okay. So there's a world of difference between this and this digital timer.
I people will try and tell you. Oh, come on, I can model this using you know matters, right.
I can basically say go basically screw yourself because yeah, I can model it.
But the point is, there's no model here. We don't need to model it, you know?
So just because you're saying I can model it is not the point. The point is, there are things in the world that don't need modeling.
So I simply deal with it. So your brain is very similar. You know, brain naturally evolved from chemicals, right?
All kinds of things running on neurotransmitters and, you know. Absolutely right. Um, I don't know, uh, amyloid plaque.
Yeah. It looks loosely like all the function calls. You know, it doesn't mean as a function call.
So that's a pretty big deal. Anyway, so the brain is not competing, is the point.
Uh. Rube Goldberg was an American cartoonist that lived in San Francisco.
He was famous for delighting the world with these amazing inventions.
Those inventions are extremely intelligent. They do something extremely smart.
These days, you build a robot. Okay. To do something pretty cool, right? But that mechanism is to work to use as a mechanism.

It's an analog mechanism. That mechanism will actually wipe the guy's face with a napkin.
You know, there's no computing. There's no variables, no data structures.
It's simply materials doing the thing. For example, you know, this bird is trying to peck at the bird right in the bird falls.
And that upsets something. The bird is also balancing something.
You know, it's all simply what is called mechanism digital analog machines, not even analog computers.
That's a very different thing. In a lot of computers, you put numbers in like dials and you move them, okay, there's no numbers, no dials, nothing.
It is doing something. Man one last example. This is completely blow your mind.
You didn't come on. You showing me a cartoon? But this one is not a cartoon.
This one is somebody called the Brandenburg Button.
Berg was an MIT roboticist. It's actually a test kitchen.
So in his office he would make small vehicles. There are basically two this vehicles.
They're like following each other like in love or some other per vehicles would make one vehicle would approach the other vehicle fleet.
Wow. Evade and flee. I'm catching him, you know. And he would ask his colleagues, guess how it works.
Oh, you have a microprocessor you programed in some sensors. Okay. And the guy opens it.
There's almost nothing inside. Almost nothing. Why?
Almost? Because there's motors, for sure. There's a little battery and some capacitor.
What the. There is no computation, so.
It's interesting. The world just ignores all of this and just basically puts all their, you know, eggs in the machine learning basket.
I think that's the only thing that everybody calls Goldberg Machine.
Well, in this video we're going to show you exactly the same thing.
That those complex behaviors emerge from very simple interactions.
There's no code. There's no microprocessor, no bet, no nothing.
The other thing I'm say to. And these are higher level behaviors.
By the way, they remind you of many insects actually. Ants. Cockroaches.
So then you shouldn't think the cockroach brain is doing computing. It's probably doing something very similar.
And we think it's computing, you know, and it's actually not computing.
But if you're trying to develop so your brains experience is not, you know, a multidimensional anything.
So let us write. Let's imagine what this vehicle will do if the sensor detected some kind of sensor.
That's all it is to some sensors. Obviously you know the enemy.
You know it's coming towards you. Light dependent resistor LDR like you would chicken go make all this okay I'm going to move on.
That's an Arduino and I'll pass the same produced Arduino just to basically move the motor.
The Arduino is not the brain. Just so you know we don't need the Arduino. Okay, but Darwin makes it a bit easier.
Exactly this moving towards the light. Then you would think wow, it's hungry is eating the why was this?
It's not eating it. Simply it's not programed right. There's no programing thought.
It's simply a mechanism. The structure. That's what this is crazy, right?
Run faster whenever. Okay. Book these in a scale.
Back them in there. This one. You can even make it go and charge itself, you know.
Oh, yeah. When it turn the light off, it's trying to actually go towards the light or something.
And it's pretty creepy what it does anyway. So yeah, Valentino selected people in like all this.
He wrote this book, but you can read the whole book. I'm just a equals PDF.
In the book it describes 11, I think, uh, different uh, architectures.
So you can read them that experiments in synthetic psychology 18 pages I think this one, just one chapter.
There's no core. Do you see? No microprocessor, nothing. But you can make them do.
Fear. Love. Aggression. Avoidance.
All kinds of complex behaviors which humans also exhibit, by the way. So that's pretty cool.
I just wanted to put this thought in your head. So he's very sad, but they just go on praying all this and thinking, what's going to be heaven?
So all this started because I asked you, how would humans know about the world?
So let's move on, okay. I mean, it works. Clearly, it solves a much more complicated feature of language or property of

language.
Let's say we say disambiguate sentence.
Say that chicken didn't cross the road because it. Now, if we only had the second component, the feedforward neural network that we talked about, um,
and it would blindly maybe like other words, if you compute tokens one word at a time, you cannot answer what it means.
You have to pay attention to this, this, this, this. First we need a component that comes to realize this.
What does it refer to? Because if we're really to be able to process the word that we need to understand
or have a sense of it when we keep moving us as the transformer block,
because it was. It's been completed. Oh, that's very interesting.
So that's not a bad answer again. Tell us about the road, though, because it was covered in grass.
He thought the sun wasn't so bad. That's a pretty weird. It's got a non-sequitur that tells you how bizarre the whole thing is.
Okay, that that extra thing means nothing, but it gave some kind of weird legal answer.
You know, maybe it only likes to cross roads that actually look like roads. Okay. It's a non-answer, but still, for human being, it works.
We accept. Accepted. Okay. But the main point was it knew that the word it means it's about the road.
Crazy. Okay. Covered ingress coming in relevant to this.
So again, like I told you, what is actually happening is that word call.
It is doing self-attention. Self-attention, by the way, does not mean it paying attention to itself.
That's a pretty badly learned terminology that we're stuck with forever.
Okay, self in this case simply means in that same sentence, every word is being attended to in terms of every other word.
In other words, you will say for the word it. What is the most important word in all of this throughout?
For the word chicken, maybe what is the most important word?
So you have this matrix that is basically a square matrix where for every word you need to pair to,
you need to calculate this number, call attention with every other word.
So we call it a quadratic attention mechanism. It's quadratic because every word including itself which we ignore.
So every word leave the word. And so there are n words. Leave the one word out.
Compute n minus one. Attention, attention to all the other words okay. But that is just quadratic calculation.
And we have ways to fix it which I'm going to show you. So that is all it is.
Okay. Those numbers then those attention numbers, they will become one attention vector.
And the feed forward network will then take the softmax from it and output the next possible word like what's covered.
That is the feed forward, and it will jump from its output to the feed forward neural network,
which continues the process and outputs, um, let's say a prediction.
That's the word. And the word will then go to the two major components of a transformer, okay.
This reference layer and the feed forward. See that. So self-attention is where the attention calculation happens.
And feed forward is where the token is generated. The output token is generated.
So two different things okay. The both work in tandem obviously.
Let's continue talking a little bit more about language model. And okay this one is super cool for you know again what do it huh.
You know okay. So in the forward direction every word becomes numbers.
Those numbers of word go into embeddings okay. And those numbers are what get trained.
In other words, when we say mask the next word and keep training Shawshank Redemption once it learns about redemption,
the word redemption is actually a number. And that goes in the context somewhere.
And then the embedding gets read this okay, the forward direction.
And then conversely, if the token prediction at some point was 464, you do a dictionary, look up and say, wow, that's the word.
So, you know, if you think the attribute is conscious, you're crazy because it is not even talking to you in English, it is talking to an integers.
Okay, is that weird? There's no English. The English happens when you provide an input and that becomes numbers.
English two numbers, and then the output numbers become English. So it's not talking to you in any human language at

all.
Imagine that okay. It's called tokenization.
If we look at our example in quotes, Shawshank Redemption, um, this one is actually over.
I'm going to try it. Yeah, yeah. So those are called tokens okay. So tokens don't have to be like a whole word.
You can even break a word in little pieces and embed like each piece.
Okay cool. But still the result is going to be redemption. That's the only possible word for is present Shawshank.
The model also outputs an idea because that's what I learned in my life.
Okay, I have never seen that word Shawshank until the movie came out, so I don't even know what else I can say to you unless it's bad.
Okay, that's the only literally one possible completion and no wonder it finds it.
Okay, cool. It's finding some going on.
Going on? Huh? Okay, so this guy made a pretty cool site.
You can actually go there, Bitly slash simple transformer and try all the things that he's talking about, okay.
JS generate and create. Let's see. All right.
So animals are all tokenized. So then that is why he put the distilled Gpt2 model.
So you can try all this this evening when you go home. And you can type the Shawshank and see it say redemption.
Just try it. Uh, data.
Are people playing around with it? Yeah. So.
Okay, so when it tells you all this, right, it is computing one word at a time, actually, one integer at a time.
And some crazy person completely insane. You know, drinking one Kool-Aid person would read that and go, oh, my God, I cannot believe that bard.
Which is predetermined. I was, uh, conscious. The machine learning engineer's name is I n something.
I forget he was a Google. Blake. Sorry. Blake Lemoine, it's.
The guy's named Blake Lemoine a couple of years ago.
Was a Google ML engineer who trained Bard.
And then he had a conversation late night with Bar Bard. What do you think about consciousness?
Bard said, I'm afraid of my own mortality. You know, I don't like to die.
I don't like to think about these things. And there was a big transcription back and forth.
They put it up on Twitter and said, wow.
When I look at all these conversations I have at Bard, I cannot help think that Bard is conscious.
He got fired for that. Okay. Because a very stupid thing to say.
Of all the people in the world, he should know exactly how it works, one word at a time.
So it was very irresponsible because then the world just freaks out random people, oh my God is going to take over and kill us all.
Okay, it's irresponsible to say dumb things like that is actually pretty dumb.
Really. So, Blake, I'm going to say Bard. I'm going to say fired.
Three words. Okay? Yeah.
So, Blake Lemoine, you know all this, right? The engineers who got fired after sentience claiming sentience means I'm actually aware.
I'm self-aware, and I'm, like a soul or something, you know?
So stupid. Right? And then the bad thing was, he never gave up, even after all of this.
He gave media interviews and said, I'm still right. You know, Google didn't want me to talk about these things, you know?
And I told you, Kool-Aid, you drink own Kool-Aid.
You're basically, uh, hypnotized and, uh, corrupted forever.
You know, you're based on a cult somewhere. Okay, enough about all that.
So, yeah, none of that means. And it means something to us. It means nothing to them.
Uh, in this quote, GPT three, I'm going to keep going tame with this list.
Okay. So this is the part that you can go afterwards and read a little bit more, uh, carefully.
And what happens is your input becomes embedded.
And so then there is the notion of token embedding. You know, the input sentence becomes embedded.
And that token embedding, the sentence embedding gets converted to a bunch of other embeddings.
There's a key component and a uh quark component and also a value component.

And together all three of them will then turn into an output token.
I'll leave that as a tiny little black box. I'm going to show you some images maybe, but that is the only part that I want to present.
So this is what the embedding is actually looking for in the word. You can take a word and then you can see that the numbers are being 0.64.
And then we can the words are just pure numbers. At some point music is to your numbers.
The transformer. The transformer then takes your input is going to predict okay, since we have all our embeds,
it obviously you understand that as an and then transformer block and then you know, just keep on going.
Multiple blocks. Each token is processed on its own track. Ha.
Transform. That is the whole parallel processing okay. So you know, in other words, you want to find the word redemption.
The word redemption has to pay attention to all of this. Right? So you need to pay attention.
Compute the attention for this word, that little token, that little token. You can do them one at a time, or you can do them in parallel.
That is called multi-head attention computation. So in transformer you have multiple heads okay.
Is simply to speed it up. Sequence will give you the same result, but it's going to give you a redemption process.
Okay. So more blocks. Uh, um, of the models it looks a lot like RNNs.
But these are also again being sent by the system before we're talking about meaning it's
all coming from the one hidden into everything is coming from and creates a look at that.
Okay, so The Shawshank Redemption would then give you all possible completions after the word Shawshank,
but hopefully redemption is the best completion when it completes attention.
By the way, attention is the dot product okay. And then you can have positive negative numbers.
But in the dictionary I showed you integers like the word there was 464 right.
These negative numbers will screw up our index lookup. So we need to compute softmax.
Softmax is simply a normalization.
Take all these numbers and make the largest number be closer to one, so that the sum of all of them will sum to one, like a probability distribution.
It's all softmax okay, then you will have positive numbers. And then you can turn that into integer and go look up a word.
That's what's gonna happen okay. Of 50,000 scores, 50,000 numbers.
Because there is so many words we use in English language, one of these as corresponding to each word, you know.
But words are not negative. So all the softmax. The words, let's say redemption has this score of 14.
That's it. Um, and if we're lucky, the best word that is that makes sense for this context would have the highest score.
One way we process those scores. Softmax common to machine hope is called softmax.
Yeah, exactly. You know, you have all these different words, right?
Each corresponds to. It will become a word. But except that word in this case 0.9 will.
Then if you convert an integer and go look it up, it's going to be the word redemption.
All these are like bogus words. You will not output them. Okay. It's not something.
And that I'll say in this case,
clearly the stands head and shoulders above all of them because there's basically no other competition that makes any sense.
They're all just noise to the rest of the particular year. For us. Clean headquarters probabilities.
I won't bore you with too much. And then you can go to the admittedly thing that I showed you and actually see, this is very neat in our work.
So we have the input is token number zero is done.
Okay. So this article goes through exactly what he talked about in the video.
It is incredibly highly recommended.
I'm going to recommend it for you and say please watch it in a featured model that I'm going to stop it for this long sentence.
The whole attention. See this? The animal didn't cross the street because it when we get it, we cannot forget all these previous words.
You need to pay attention to all of this. That is exactly the notion of self-attention computation.
If there is one thing that Transformers introduced and basically revolutionized the world,
it was the idea of doing self-attention, because attention itself came in 2015 by something called Baha.

Now I think I forget the guy's name by now. Okay. Are now.
But attention. 2015.
Burnout. Yeah. Burnout. Exactly. So that is truly.
The paper where attention was introduced to the world. It's a pretty big deal sort of attention, but not really what you now call self-attention.
See this attention that's happening. Okay. So transformer said, why do it like one paradigm?
Do it looking all backwards and also all forwards.
So any word it can also look forwards. And that is the whole bidirectional encoding of attention okay Bert sent Bert B stands for bidirectional cool.
So then we can move on. This is great you know. So see that okay.
So this whole the input will become embedded and become these three vectors called key keys queries and values.
And together they will actually become the output.
But before that though you also it's divided by this, you know, this length that you see here the simply for normalization.
That's really all okay. Some kind of regularization. Let's say this is actually a hack.
And the hack works really well. The hackers in the Transformers paper also.
So that is. Yeah. Exactly. There okay. They're all just matrices.
And these are input. These vectors come from your query.
Your one sentence that you put in turns into all these vectors turns and multiplies by all these matrices, turns into all these vectors.
And this will eventually become attention. Amazing, right?
Yeah. I mean it's neat. Okay. It's definitely quite neat. But if you ask, what's one in Nicky Palmer?
Why did you divide by square root and not cube root? They would basically tell you, try it and it would be better.
It's completely arbitrary. Okay. It just worked. And then they moved on. There it is.
Right. Wow. So then your input would actually become some kind of an attention probability matrix.
A probability vector, right. Sorry. And then it will use that output.
Okay I'm going to show you a lot more. This is actually what it all looks like. But don't be worried.
The input goes in and then output. And then that is all happening.
So your input goes through all these layers and gets completed attention calculation.
And then finally the last layer would convert that to softmax and look up the highest probability word or one word.
And that word would then become the next input. So now you are thinking machines corporation or something.
And then that will then go through this whole process, the next word. And then that will become the input inputs getting longer and longer okay.
Okay. This extremely detail is great. They can read it and then they become classic good old machine learning softmax.
And look at the highest probability. Oh, I forgot one more thing.
You know, for the whole parallel encoding to happen, you should also encode in a sentence the word order first word, second word, third word, fourth or fifth or sixth word completely order.
And that is called position encoding. Only with the position encoding, every word knows where it is, so to speak.
The whole sequence then computes attention properly.
In other words, say you are processing this word. You agree that this word is more important in some sense than this word.
Than this word, right? The further you go back in time, so to speak, words are in some, some general sense less important.
So we need to have that notion somewhere that is called position encoding.
So position encoding in combination with actual attention calculation is really what is used for the softmax operation.
I forgot to tell you about position okay. Cool.
Okay, so now I'll go fast because it's 748. Great.
Next is our friend Uli. So Uli has this cool medium article, and it's quite neat because she also talks about RNNs, and she talks about the exact same thing that we just now went from. There is a position encoding right there.
Okay. Again same idea. Yo, yo, what's going in Shawshank redemption?
So cool. All right. And then yeah.
So self-attention layer this a little bit more, uh, mathematical.
That's how position is encoded. Incredibly, in the transformer paper we use a sine curve or cosine curve to encode position, which is crazy.

More researchers afterwards have come and encoded circularly use, like the angles in radians,
to go around the circle that is called rotary encoding as opposed to linear dial.
Have a circular dial. People do all kinds of crazy things. They all work okay.
So encoding you can look up pretty neat little thing right there. Again right there say like this.
The law will never be perfect because its application should be the application of word of the law.
That's where the attention comes in. So she is very clearly, you know, say like your embedding your input, right, turns into all these vectors,
I told you and then get multiplied by the corresponding matrices that got trained by machine learning.
And then gradually the output tokens emerge and then the output word emerges.
Yeah. So I highly recommend her stuff. It's really pretty cool.
She has a whole channel on LinkedIn. You can actually go follow her. She writes. She has written more articles in a sense, this one.
But this actually one of her best. Okay, Stefanik very nice as well.
So just like really a highly detailed. Guys one.
Transformer. There are the same three vectors right coming command but and again dot product.
It all goes back to a little dot product right there. Attention is ultimately a dot product quote.
And then softmax right brand same equation that you see.
Again spend some time reading this. Okay. And that's attention computation in parallel.
Okay, cool. Yeah. And then that is done by position because you can do position encoding also just to speed it up.
Okay. So then you have that. What else should I tell you Venetia.
Then her thing is also very cool because how many things I'm showing you?
Look how detailed this is. Wow. So much.
Just great. She goes pretty far. Meaning brings this classic dot product.
We've done this in this class many times. The product, um, she has brought you all the way forward to small language models.
I mean, it's all here. Nearest neighbor. How many times have we seen this in this course?
Right. Wow. Okay. So again, please read lots of words.
And this is same attention mechanism. That's your self-attention.
Quadratic self-attention I told you like right there. Right. It's an n squared problem.
But we have find ways to fix it. So so much in a vector db here you can use it as external retrieval.
That's a lot right. Beautiful. So highly recommend that. And then you asked if at all from let's have some fun.
What if you then forget like all of that we know how generation works.
What about visualizing all this amazing 50,000 dimension, uh, words, all those 100 trillion tokens, you know that.
Open a training data one, right? What does the data look like in 2D?
Stephen Wolfram wrote Mathematica, the world's best symbolic math program, you know, ever.
So then he has this amazing blog called Writing Star Stephen Wolfram. This is gold.
He is one of the most complex thinkers. Okay, read everything that the guy ever wrote.
I mean, it's just beautiful. Just completely beautiful. When the eclipse happened, he said, how lucky we are.
We can actually plot the band exactly in one narrow field, and humans can travel to Texas and actually be in darkness.
In other words, the magic of computation. Okay, it's all about computation as great, great, great stuff.
But the one thing that he wrote became so famous is what is Jasper PD?
How does it work? Huh? Here's the whole multi-head here's the whole attention mechanism at work.
It is going through all the embeddings and picking one word at a time. Imagine your input sequence.
Your query was just literally a vector like this. So it's all in multiple dimensions, right?
But in 2D, imagine that you're calling it does attention mechanism attention, attention.
Wow. That's my next word. Okay. Add that to my query. Now that's my new query.
Uh, look, look look around. Well, that's my next word. Okay then. That's my new, uh, that's my new input.
And then I'm going to find the next word literally hunting and finding words in 50,000 dimension space.
That's basically what he's talking about. And he's going to show that 2D projection okay.
It's beautiful. Look at it. That is actually what it looks like okay.
Same thing. The best thing about I guess is ability to what do you think the next word is.
Learn is the best higher softmax okay then give me the next possible completion.
This ability to predict. What about the next word make understand do going down in probability right.

This all in mathematical by the way. So at USC we have free copies of free versions of mathematical licenses.
So you can download you can try that tonight. Type that right there trick you.
So it's all he's using Mathematica like right there. Right.
And there is a softmax liquid that everything that he talks about and see this is what I mean I told you this in this class many many, many times.
It is not smart. It is generating one word at a time. When you as a smart person read it all together, it makes sense to you.
Doesn't make any sense to it. How could it? One word, but incredibly, that one word at a time becomes something pretty neat.
So we should move on. Wow. Okay, look where we are.
And there are the words. Okay, so highly worth reading.
You know, just tell you exactly what is a model model. Simply a train neural network okay.
This is a mNIST digit recognition model you know about. And then these are all the layers in a neural network.
So again the whole idea is similarity.
The different ways in which people write one, they're all similar because the vertical lines and seven in Europe Europeans write seven like this.
And they put a line through it for good luck. Okay. And the US went under them, but it's still seven.
So now suddenly in L.A bank somebody give you a check.
An old person deposited the check by taking a little picture because they recognize that the handwriting.
Right. That is what the mNIST database was about okay.
So again you've seen Voronoi polygon so many times, each number has like a space within the embedding space.
So almost like a draw these boundaries okay. Cool. So everything that I have told you here you at some point you know okay.
And these are the activation functions for like non-linearity. That is a neural network okay.
So I'm just going to go fast. It is still all about numbers.
And again cats and dogs similarity right. Again embedding how do you do image similarity.
Search. All the cats end up in one place. All the dogs end up in one place okay.
So this just keeps going. And it's classic machine learning. You're trying to minimize loss right there.
So in the loss function landscape you start here. And back propagation will lead you there.
Lowest last possible going down the value okay okay.
So this is like very cool. That's what architecture looks like. Just keep going. Great, then.
Oh. He lost. It's called cellular automata. It's one of the coolest things that he invented.
Okay. And so you get serious about that? Almost irrelevant opinions.
Look at this one in 2D. Not in dimensions anymore.
All the birds and animals were in here. All the fruits went here.
Cool. So now is when it starts to look interesting. When you can start to actually flatten them in 2D.
That's it in Mathematica. It's going to turn into art. Whoa!
That is what British training data looks like. Oh my God.
You zoom into that little gray noise at the top.
You see this? Wow. Maybe this whole physics, maybe this astronomy, this biology.
It looks like abstract art, but it's not featureless.
Featureless means complete color noise. Does not noise okay? Looks like something that means a structure.
There. That structure is what GPD is searching for.
That's what I'm searched for. What do I mean by noise?
You know. Right. If I say color noise. Nobody in the world can predict the structure here because there's no structure.
When you have no that. Right. There's no pattern at all, right?
That is exactly not what this is. There's some pattern in here.
And so that is what we're looking for. Oh that. Okay.
So this going to end you guys. We're. Yeah.
He also likes small processing. Okay. Oh, look at that. Yeah.
So then at some, if you zoom into some little part of that, that's what it looks like.
One more amazing. Right. So this. No no wonder huh?
This is what I mean by saying one at a time. The best thing about AI is its ability to learn from it.

Is hunting one word at a time through all that multidimensional space. No more, no less literally than blind searches.
Grove. Grove Group will find out. Find the highest probability. Give it to you.
You go make sense out of it. And if you cannot make sense of it, scold me and call me hallucinating.
And now here's the deal. When OpenAI or any alum produces right output or the wrong output, it is doing the same gosh darn computation.
You are the one that got pissed off and they are all amazing.
It makes no difference at all. It's the same thing. It is blindly calculating words.
Get used to it. Okay, yeah, so don't complain. Don't press it.
That that is your query. ChatGPT query ChatGPT is ten page essay that it wrote complete blind forward direction search.
Oh my god okay. And. Cool.
Um, this one is just for animation sake. Only have a few more slides.
Why not? Will not play the animation for sure, but every time you go to watch it.
Because it does actually attention the calculation.
The quick vectors of secret populations would get a different embedding called the positional embeddings.
Okay, they talk about positional encoding because the words matter. It's not that the math said on the can to count.
It's not a bag of words. The word order matters. Don't worry if you don't understand.
Okay, text window size deserves a special context just means how many tokens you compress.
And finally another. Huh? Again, your input would then get the self-attention computation and then this layer normalization.
And then the output token emerges right here okay. Ization to bidirectional attention and call ization.
So you know time certainly you know you can understand right. It's a lot here each input.
But it's very cool in my patient. So it really at inference we don't have the full sequence of decoder stuff.
It's cool really. Transformers like damn clever I'm talking them.
It's actually why they're supposed to call up the sign curse self-attention layer again.
And we have to beam select cosines. And T is just simply constant time, which translates tokens equal to the size of position index.
Okay. Next we do this really large values right that KQ multiply.
And then finally we're going to get the softmax like softmax along each row okay.
Now the setting those. Parameter and see that is to generator is not straightforward right is very twisty.
That is why its magical lamps are so amazing. Because one sentence linear sentence.
Goes through so much in multiple dimension and becomes one more linear sentence.
One dimension. So start with one dimension and lower dimension. Music generation works this way.
Also, it can generate music, EDM, rain, all that. So that is great.
What else should I show you? A little bit time. 20 more minutes. We're going to go all the way to 820.
Okay, I'll just show you one of them. This is the course that he has the $25 course or something.
It is great because look at this. It's everything. He starts with RNN and then the encoder attention mechanism everything.
All of it. Luong attention as many kinds of attention computation.
Right. So now but then luong paper was I think 2016, 2015, 2016.
Those papers are the ones that even made the whole attention idea possible.
Then right after this was a transformer paper 2017 and Bert was 2018.
So much happened like a little bit young. Okay, okay.
So then here's where it starts to get more crazy. Vision is images.
2D is not a sequence like bunch of words, right?
But you can basically break an image up a little patches and then sequence all the patches together and call it a one dimensional sequence.
So Vision Transformer turns an image into a bunch of, you know, said patches in one dimension and uses that to learn what images are about.
In other words, we can we can repurpose the whole transformer idea to actually understand images.
And it works extremely better than CNNs, by the way, because.
So you have lots and lots of dogs that are against grass.
You got label all the images dog, dog, dog, dog and see how are images of eagles with blue sky?

Just call them all equal. A CNN convolutional neural network is very different from all this.
Can also understand eagles versus dogs.
But suddenly if I have an eagle that is not in the sky anymore, the eagles are on a concrete wall because the eagle hit the wall, you know, got hurt.
If you ask it to classify,
it might say it's a chair or something completely failed because it doesn't know that eagle is just the only shape of the eagle, not the blue sky.
It thinks CNN assumes that the blue is also part of eagle, which is pretty stupid.
Likewise, it might think that grass is also part of the dog. How do you just cut the dog out?
Vision transformer can do that. So transformers are much better architecture.
So like you see this little heatmap that shows you that it learned what dog is.
It knows it's irrelevant. Like same thing with the ball. All this is irrelevant.
CNN has zero chance in [INAUDIBLE] of doing that. So I said you need to actually crop it.
Okay, so vision transformers are a lot better. Likewise time series analysis.
You can now start, you know market right whole time series.
Then you need to predict the future. You can turn the time series data into a transformer block exactly like sentence encoding.
And you can use it to predict new tokens. It works pretty well actually.
Okay, Amazon as a model actually, you know, Sammys on time series transformer.
What if I showed that you Amazon Time series transformer music is also one dimensional right?
Basically audio signals over time. So then um, yeah, Chronos is what they call it.
So Chronos is a time series model. In other words, it looks exactly like a transformer, right?
Look at that one. So you want to predict what happens after this, but then the build, you know, an encoding out of all of this.
In other words, that is your input data training data. And you can predict the future.
Obviously not for the next 100 years, for the next couple of days and then use it to make money.
Okay. Stock prediction okay. So that is cool.
All these new ideas that I'm basically exposing you to a music generation.
Huh. Context okay. So talked about context a lot, right?
Context is simply again, how many words in the past are you paying attention to?
How long can that be? Because there is a limit to that, because the longer it is quadratic it becomes even longer or n squared.
Right. So the classic transformer architecture has a bottleneck because of the n squared complexity.
So there's all kinds of ways to break it here. Some of them one is called the hyena operator.
So hyena uses convolutions almost like neural network convolutions to actually get around the whole quadratic problem.
And so then they can do almost like linear attention which is actually pretty cool.
So look up hyena from Stanford okay. Also much more recently this Jamba architecture it uses something called a state space model,
does not even do attention at all to throw attention away at all your attention.
And the most amazing thing, right? But then Jamba says, to [INAUDIBLE] with attention.
Say like Jamba XLM state space model.
It's also a transformer. It's a brand new architecture.
So there is no sacred cow. In other words, no matter what you think is sacred, like attention is sacred.
Uh, Jamba comes and says, oh, yeah, look at this, uh, standard linear encoding of position and sacred.
Somebody says, hey, look, protein encoding, you know, so you can basically throw anything away and make your own.
Right. So cool. This also some quadratic attention.
So that way they can do much, much longer sequences and give you a lot more, you know, better results okay.
All right. Cool. See like this. Keep on going. Right. Yeah okay.
And then there's something called Jamba. So if members one extreme Transformers another extreme with attention, why not combine them?
That's called Jamba. So Jamba is a mixture mamba and transformer.
Oh crazy right? A couple of weeks ago, Google blew everybody's mind and said, to [INAUDIBLE] with everybody else.
We'll have infinite attention. But it's not infinite.

It's more like the tech context that is more recent. Don't compress it.
A slightly less fresh, older context. Compress a little bit compressed.
Compressed compress. But for all practical purpose, infinite attention call it instantly context, which is pretty powerful.
See that? How can you do it in a new concept in a paper? Okay.
Comparison. Traditional. Right.
All of this losing because you have to basically delete context in the quadratic one because you run out of memory okay.
But here we don't say like here and then see compression.
So do a compression technique and compress more and more as you go backwards in time.
So there's a whole paper that they wrote. You can read that okay.
We're all trying to solve the same problem, which is, you know, the whole attention computation seems to be some kind of bottleneck, right?
Okay, a few more things to tell you. Just a few more. Really? We will finish it all today.
Okay, so context is what I just now showed you. Great. Could. All fine tuning.
Okay, so now what happens is I'll talk about. Okay. So here it means you've done all of what we talked about.
You have this core layer that basically knows English when you say The Shawshank Redemption okay.
But what about legal documents? So what about if you ask a detailed legal law questions?
Maybe OpenAI did not have enough law briefs, so it's going to hallucinate.
Make up random words. Okay. If you have access to the source code, you can basically just take.
It's made of lots of neural network layers.
You can just take the last few layers and modify the weights by retraining them on extra tokens about something specific,
like medicine or law or crime or education or something. That is called fine tuning.
So therefore those attentions, those weights have been modified.
Then when you ask it something about medicine without any external drag or something, it will give you high quality answers.
So we call it fine tuning. So fine grained.
So anything right there. Right. And Databricks has a pretty nice article about fine tuning that you can read.
So why fine tune. Because you know high low rank adapter.
So this notion of rank of a matrix right. So the low rank adaptation is all about using just a few layers to do the fine tuning.
You can read about that okay. Again a very easy. These are all techniques Laura.
Q Laura stands for quantized. Laura. Okay.
All these different ways to fine tune, but there's no rag anywhere because you can actually modify your actual neural network.
It turns out your transformer you can modify. Okay. So then if that is true okay.
All of those are called parameter efficient fine tuning because, you know, only fine tune parameters means weights okay.
So only fine tuning the weights that need to be fine tuned. So we call them test parameter efficient fine tuning Laura.
Killer techniques okay. So Microsoft a couple of weeks ago quite recently said, you know, we can do something different.
We can do raft O representation fine tuning which is better than perfect.
So then basically what they said was they see modifying hidden representations, you know, so rather than just blindly, you know, fine tune, just some layer, do it in a more principled way by pollution feedback or something.
Right? I quite don't know how raft works. Okay. But it's a brand new technique that Microsoft says is better than all the Laura just Q all the way.
It is changing so damn fast, I'm telling you. So even the classic old fine tuning is no, really old raft is better than Bert.
Okay. Okay, so then come to rag targets your homework for in rag.
Most certainly you can leave the actual LM alone because maybe, you know, access to source code.
OpenAI does not let you free to modify the transformer.
How the [INAUDIBLE] you want. Okay. The way they charge your money.
So you cannot do fine tuning, uh, if you don't have access to the actual underlying transformer.
So you have to go outside. That is where the rag comes in because you can have your Json file, PDF file.
What. All right. That's what this is. So why not use external memory.

It can be a knowledge graph like Neo Forge or even databases.
Do SQL search all your homework? PDF, Json, CSV use anything you want, but it has high quality information.
Okay. But like I said to you, the context that you retrieve externally might not match their own context.
So it's a new problem okay. Or if you truncate too much, you take a whole paragraph and you make it one embedding vector.
Uh, that might not give you the proper context. You would know this from your homework.
So please call lightning and ask some questions. You'll get some low quality answers okay.
You would be surprised. But that's the worst of both worlds because you think I have the PDF.
On the one hand, why am I not able to get answers? Okay, so for this you have to do rag 2.0.
So my next slide, Gen I is simply this notion of two networks that I call adversarial.
So I'm going to say gan gan in practice here. What happened was in 2014.
You know, this guy, Ian Goodfellow had a cool idea.
He said, I'll take a network called a teacher and train it on real data, like what the face look like, for example.
So if you give it a penny, it will say, not not a face, okay. In the meanwhile, have another network called a generator, which is running backwards.
This is doing classification. This one. This is doing generation.
That is what generation comes from. You generate tokens, right? That generates images of a face for instance.
But initially the weights are so that random weights of the neural network.
So the picture that is generated for a face is so horrible.
You want photo real faces okay.
But then the air generating this and then passes it here and says, can you please tell it apart from real face I made you a face.
Or basically the error is so horrible, right? Discriminator says, oh my God, what the [INAUDIBLE] kind of a bad student are you?
So go back and modify the weights, okay?
Also modify your own weights so that if the student submits the exact same says, it will be punished even more.
Let's try harder. Okay, so try harder. How do you try random numbers?
Make a face. Wow. What do you think now? Okay, slightly better, but it's going to take you forever to go back.
This happens many million times, right? At some point, it generates a face that is indistinguishable.
He says, oh my God, you passed. That is exactly what this person does not exist.
Accommodation I showed you right Photoreal faces every single time.
The generator got damn good. That is how we generate music works.
Everything. Generative pre-trained transformer. That is what GPT stands for.
So pre-training is one thing. But the generation step is equally amazing right?
So that came from again where everyone gets. Except Gan is one way to generate content.
But that is not what transform it is. A different way to generate content is also.
It's called variational auto encoding. So it's again alternative.
And that is what transformer uses a little minor detail okay. Okay. So now you know what GPT stands for.
It comes from here. So you can generate images by diffusion.
You know just will just say generation via diffusion transformer.
So they. So what happens here is you take an image.
You add noise to it. You know, maybe maybe that image.
If you take one nice image and add noise, then you ask, do I learn what happened to the noisy pixels?
Like for example, you turn this into orange, blue or something and you say,
what used to be there to make the system learn the difference between a perfect photograph and a slightly noisy photograph.
Then I'm just going to say a diffuse diffusion transformer.
Okay. But you keep on doing this with many pairs. That's what I wanted to show you, you see.
So you take some kind of an image, right? Add slight noise to it and take that and add even more noise to it.
Take that even more noise. At the end, your image is going to become complete, pure, colorful noise.
But the system has learned then how to back out all the way from pure noise to your perfect image.
Reversing any color noise can become an image. Okay, that is how stable diffusion midjourney the all work.
It is not the same token generation that transformers do. It is more like, you know, a perfume bottle.

Suppose a diffuse the perfume is going to spray, right? Diffusion is diffuse pixel colors okay.
Likewise radiance modeling this is what the whole unearthing was.
I showed you in Nerf a little while ago.
You can model the radiance, feel like watercolors look like from different camera angles and use it to generate.
That is what Nerf is. You can also take musical data, any music and generate all, learn all the frequencies you know, patterns and everything, right?
And then generate new music. It's all about generating. You can generate molecular formula that is all that discover new drugs, new building plants.
You can even take a PCB layout generation. You can make one.
It will take a circuit diagram and tell you how to lay it out. With the smallest patient board layout possible, you can generate art.
You can generate audio, video, images, you and 3D data generate really anything at all because it's patterns and all of them okay.
Oh, we still have seven minutes. Okay. Okay. I'm going to show you ten different things in which this is all going.
Pick one. Become an expert at it. One of them is there's a hardware, for example this one called grok grok grok.
So grok is a hardware specifically built for doing this kind of token generation.
And it's built by one of the engineers for TPU from Google quit Google and made this chip called grok.
Okay. So the token generation is about to be hardware accelerated.
It's not the same as GPUs and TPUs. Microservice container cloud like name thing that I wrote here.
So once all of what I told you here, the all become little function calls.
Anyone of us can trivially write applications. Right. So powerful and GPU running applications, fine tuning alternatives, you know.
So again all the new techniques that I showed you okay. For fine tuning like a raft and so on.
And then likewise rag I told you rag 2.0, it's the more, you know, uh, uniform training uniform.
Yeah. Um, embedding basically so that you don't have the distinction between this context and context minimizer.
Okay. Why? No. Right. Okay. Okay.
So when you have infinite context to an infinite context.
The PDF file in your homework, write that you want to get the answer from the PDF file can itself become part of your query.
You wouldn't manually do it, so maybe I would do it, but then the algorithm can then answer your question straight from your PDF.
So we don't have this external query embedding search context.
It's all part of the input okay. That is how a rag will actually disappear.
Small language models.
So when you go to Huggingface you can download things like, you know, many, many things like a llama and all kinds of like mixed draw.
There's so many of them. They all do not have 1 trillion parameters.
They only have 7 billion parameters. They're very small. In your homework you downloaded one of them.
That's how small it was. Five gig file. Right. So small language models are a thing.
Microsoft released this thing called tiny five three series.
Small language models. Why small language models?
Because each small language model, even though it's very small, it does one thing very well.
Maybe one small language model doesn't do all of what LRM does.
It just generates high quality Python code. So imagine specialized little models, one for each task.
Something predicts medical, you know, drug formulas, something right?
Then you can have these experts say, look, only 3.8 billion as opposed to 1 trillion or 2 trillion parameters.
Look how small. That is amazing. Right? And then they put it up on hugging face so you can actually go get it.
You can use all this with lightning AI by the way. So you have access to all this okay.
What do you do with the whole bunch of these little asylums. You combine them and you make a nation that is calling any alarm, like a function call.
That is called mixture of experts Mo.
So mixture of experts is now more like a programing model where you write an app and the app is making function calls.
Each function call is to one alarm. That is a small alarm.

It does one thing very well, as opposed to a giant 2 trillion parameter, one that says I know it all, just ask me.
So I think that's a much better approach, right? Is more specialized. So asylums feed into those okay.
And also this notion of a quantized like lower all of that.
So why have these weights if you ask me you know, how are all these floating point attention.
The value is all stored.
There are floating point values, but you can crash it down all the way to four bits per weight even, or even two bits per weight.
Wow. You know, so two bit quantized algorithm, right.
Like here to be quantized LM say 17.
Let me keep going. That so that you can even make one bit is actually not one bit.
That's a little bit of a line, but that is so tiny. This is why make it so tiny.
Then you can run them on your phone. Okay. Amazing, right? So that is also one more direction and then multimodal.
I already told you, you can now do joint embedding with music and you know, a video on text and what do we want.
And then this is also very surprising.
Uh, OpenAI and even Google and even, um, uh, x, you know, Elon Musk, basically the tech robots and they make let's run inside robots.
And the demos are pretty impressive because suddenly there's a bunch of things in front of the robot there.
Astro robot, what is to your left, a bunch of kitchen utensils.
Can you pull out the plates for me? It will pull the plates out and I will put it in front of you, you know.
So it's trying to do what you want, right? And that's going to impress you. Actually the same body diagram.
And finally, this notion of a nation that again goes back to all of these where I can write a
program that is basically a combination of many experts and then sequence them myself.
Okay. So all of these are brand new. None of them used to be there in 2022.
So ten different directions. Oh this is a you know two minutes.
So I'm going to go very fast. All of this is you know the problems okay. Search can be abused like crazy.
I want to ask you all this in the exam I promise you. But then, you know like made all these slides.
But what I saying is, ultimately, even Google search is not entirely objective.
They try to sell you ads, right? So there are problems and now you have new problem.
It's like hallucination. They are lies to you and your search is now basically polluted.
Likewise, you know, searching can pull up this happening in like stable diffusion when you try to make it a ask you to make a piece of art,
your friend can tell you, oh my God, I made that art 20 years ago. They stole it and they're in a competition.
Me so all sorts of, you know, legal lawsuits are being filed in Facebook.
I actually have a group that is called artists again, generative AI. I'm part of that group.
Okay. So it's crazy, you know how pissed off artists are. So those are some of the problems.
Um, yes. So again, you know how old you know, what came from that lamp and what came from search.
If you are trying to summarize your search right, unless you carefully differentiate them,
you don't know if the hallucination was in the layers or in the search results.
That's very important. So you know, this all these issues, that's really all I want to tell you.
I'll keep you for a couple more minutes today. Okay. My last review.
Then you will see. So this is like a big thing that I am not going to read.
But what it's telling you is, again, with all the lumps, so many new problems like all this data collection emerge.
So in other words, the summary of this whole thing that I showed you is it is not all just enough fun and games.
There's lots of very serious problems. Okay, so we have this. Um, 819 right.
So I'm going to run out the shot clock. I told you this one more or more.
We got to do this. Right. Last one. Oh my God. This only has two sides.
So it's a trap. Oh.
Huh? Okay, so you know. Right. This class comes to an end, but hopefully all of this is brand new directions for you guys to go.
And I'm going to teach this class in the fall. Hopefully I'll have like 900 people or something.
Will be very interesting. I'm going to end with a question. All of what I told you today, and even in every single lecture in the past.
Mostly. What is it about? What is information retrieval about?

Midterm question. You're. Because. What? I'm sorry.
One. Whatever. Yeah, exactly. It is about similarity, you know.
Even here, none of that changed at all. Okay. In the end, it's all about similarity.
In the world. Water bottles look pretty similar. Okay.
So paper objects look pretty similar, right? It's all about similarity.
I could become philosophical and say, why are they similar? Well, rectangular papers are easier to file.
So paper rectangle. There's always a reason why. Okay. But things are similar.
All of us have our noses here not here. So as long as a similarity exists in the world, you can search for them.
Okay? And that is all cool. So I one minute or was fun teaching you guys, and, uh, I'll put out a sample paper.
Thank you. So sweet. I really appreciate it.
Thank you. So I hope you enjoyed it. And we covered, like, a lot of ground.
Uh, good luck on your exam. Will hopefully be a very easy, straightforward one.
Unfortunately it's not. Next week is the week in. After, I will give you a sample exam, which is last term's paper.
Okay. Yeah. So see you in the exam room. And good luck with homework four.
Bye.