hey everyone thank you for watching Star morph where we talk about artificial intelligence and web development today I'm really excited to introduce you guys to Jerry who is the founder of llama index a tool to connect large language models to external data so thank you very much Jerry for coming on today thank you Dolla for having me super excited to be here absolutely
um so yeah before we kind of dive into what is llama index and how people can start using it and building with it uh would you be okay with giving a little background about what you how you got into building this tool in the first place

Jerrys background

yeah that sounds great
um so I started building the tool back in November I've always been uh very interested in AI you know systems uh startups all those types of things and so my previous work experience included um spending a few years at a series a series B startup called robust intelligence where I was leading the ml monitoring team and then before that I spent some time in AI research at Uber atg and I also spent some time for that as a machine learning engineer at quora uh so you know sometime in like machine learning across like recommendation systems like computer vision a little bit of MLP as well and so I hadn't really like fully dove into the like whole gbt3 like uh wave like around that

time like dolly was there like stable diffusion was out and like people were playing around with building the applications on top of GPT but I was just starting to get into that and I think one of the first things I ran into as I was thinking about building these applications was I was trying to feed uh this you know amazing piece of technology my own private data uh and so you know I was trying to feed it uh our customer conversations for my previous company and then I was trying to figure out how to best uh leverage this tool to be able to synthesize insights from the private data and I figured a lot of the abstractions that I was building were probably things that everybody building applications uh would need to solve anyways and that kind of motivated me to start a very initial Tool uh and at the time it was called GPT index and I've since renamed it to Lava index but the idea of the tool was basically to store some sort of like external memory for the language model
in some sort of data structure that the language model could be able to like look up and reference this memory for for later uh and so you know it very much started as a design project really but it seemed like a structure chord with uh like people people that were also trying to build applications and from there evolved pretty quickly and now it's kind of like a full-fledged toolkit that people are using to actually build an end user-facing

applications
oh it's amazing it reminds me I've heard
on Y combinator and things like that
that's Sometimes the best way to find a
real product Market fit and solution is
by building something that you need
right and yeah it turns out that I think
a lot of people are looking for exactly
what you just described the ability to
bring in their specific data into these
Bots and use it with you know embeddings
and Vector storage and and Link large
language models too
um yeah that sounds exactly and and just
to give it a little bit of background I
think you know in the beginning uh it
was almost like a design project and so
if uh I I had to give a little bit of
intuition for what this design project
consisted of is basically this idea that
a language model is basically you can
treat as like a reasoning engine
um because you give it some natural
language input and it'll give you
maximum output uh because it can uh try
to give you back the best answer given
what you tell it to do
um in that sense it's also some sort of
like AI enabled CPU right and then if
you think about the rest of the computer
architecture that you need to build a
full program you're going to need some
sort of like Ram or hardness space to
actually store stuff right that you can
basically have the CPU interact with and
so for me I was very interested in
building some sort of data structure
where the language model can actually

reference some sort of like external memory or hard disk space if you will so that I could leverage you know it could try to look up information that wasn't uh previously in its own knowledge base and try write a try to retrieve that for the user depending on the query that they ask so anyways it was this like big design project and uh I think I think as users became more interested in it I had this moment where I thought oh hey like I should actually use you know a lot of the existing techniques um and not try to reinvent the wheel and actually just try it like faking a lot of the existing stuff into this toolkit that people people could easily use to again uh store and
um use their private data with a language model awesome yeah that's

LlamaIndex Framework

really interesting so many parts that are coming out right now that are you know I think the integration and and piecing all of it together into kind of like you're saying a full stack uh AI operating system or something like that is really exciting to see
um exactly the development of how does it outlined in the framework of llama index these different pieces
yeah so I would say uh I actually don't know if there's a proper computer analogy but I think um at its Basics you can think of it as just a really good tool to just manage and query your data um and and use it with a language model

the idea here is that um the key problem
that I want to outline is let's say
you're just an individual or you know
you're an Enterprise and so you either
have like a collection of notes on your
hard drive right or you have an entire
data like with just like thousands of uh
or like gigabytes of terabytes of
different types of data like structured
data unstructured data and regardless of
whether you're on either one of these
spectrums you want to figure out
um okay like this language model
technology is amazing how do I use this
on top of data that is private to me
that it inherently doesn't know about
right and and so just like kind of
taking a little bit of a setback
um these days there's basically kind of
like two camps for figuring out how to
make knowledge into
um into the language model
the first is this idea of like
fine-tuning or training or distillation
where you know you are a machine
learning engineer or researcher that
knows how to build these models you
actually you know initiate some sort of
training process on top of this data so
that the model itself can actually just
memorize stuff uh right it can like
memorize the data that you give it and
so that's basically how you train one of
these large language models in the first
place like if you look at tragedy and
you ask it anything that's like publicly
available on the internet like any
Wikipedia page or like ask about apples

about the American Revolution like any
of those things
they'll be able to know the answer right
because that stuff's like all kind of
publicly available it was in the
training data but for your own private
data you can also initiate a process
like called fine-tuning where you can
actually fine-tune these models on top
of you know your your own private data
uh and and these days there's like
downsides to doing that um a lot of uh
fine-tuning effort
s don't quite work uh a lot of people
don't really know how to do fine tuning
you have to prepare your data in a
certain way there's a certain like kind
of expertise level required to do that
so the next uh kind of thing the next
camp that people are uh kind of like
um going into in order to you know boost
their language model data is uh this
idea of in context learning or uh like
retrieval augmented generation
you fix the language model itself so you
just use it out of the box so then you
couple it with like a retrieval model on
top of your data and basically build
this entire system where you know you're
just creating this language model as
like a processor and you're coupling it
with
um some other thing that's able to go
and retrieve your data and then put that
data into the info prompts and so
basically just a high level you know
language model is very good at reasoning
uh and and so what what you can tell the

language model is essentially hey here are some new data that you haven't seen before so here's some context
um and you don't click on the text and then given this context here's a question and then you give it a question and then given this context plus the question please give me uh an answer right and you just you can tell like track GPT or any language model this entire sequence of text and then it can like try to figure out uh that they answer for you and so you can actually use it as a strategy to operate on top of your own private data and that really is what uh like llama index is designed for yeah it's amazing I mean I've been

Data Types

I've been talking to I've been doing this now and I spend at least an hour a day talking to one of my bots that is trained on my data and I'm constantly thinking of oh what do I want to learn this week what should I add you know into the embedding to be able to learn on so I think it's
it's a really powerful way of learning and creating content and you know querying your data
um so could you talk a little bit more about that was a great high level overview on bringing in your data what about
um I know one question that a lot of people have is what kind of data do they need to put into it so you talked about unstructured versus structured data

people ask me a lot what file types
um should I you know what format does
the document need to be in or you know
these areas of how do I start to get my
data into a system like this
so that's a great question and honestly
I could talk for like an entire hour
about this but just I I hope to like
distill some of this uh down to the
users there's something very beautiful
about language models and that they can
basically just take in
um unstructured text as like a central
medium of data representation and they
can understand unstructured text very
well
um and so just to give you an example if
you take a web page right and and maybe
just do some basic processing like you
trip out the HTML tags or something like
that right and then you just like dump
the entire web page into the input
prompt of the language model you know
assuming you obey the prop limitations
but this is just an example anyways
um it can understand that web page like
the entire thing even if there's like
messy formatting and other stuff going
on in that thing that you give it and so
you don't really need to write a ton of
like manual person so like process or
clean the data
um and and like uh if you look at kind
of before language models like how you
set up like a data pipeline you need to
do a lot of conflicts like ingestion
parsing ETL and then you store it maybe
in like a structured format then you

annotate this like unstructured data
with a bunch of tags and then you use
these tags that help like gain insights
of your data one of the cool things
about language models is you just give
it unstructured data it can like just
process the entire thing like true it up
and then just figure out basically what
uh what the answer is
um and so if you have something that
powerful it kind of implies certain
things about like the setup that you're
going to use to basically uh and the
types like file formats that you can
address because you can pretty much
ingest like any file format like PDFs
PowerPoints Excel sheets anything as
long as you can just convert it into a a
text representation an unstructured text
representation that the language model
kind of understand uh this includes like
images or videos too like if you have
images and you can caption that image
using some piece of text or you know you
have a video yeah and you have a
description of that video using a piece
of text then you could totally just like
index the video and ask questions over
that video too
um and so uh What uh llama index
actually supports is we have the site um
so so just just to give you a sign we
have three different components one is
just uh data connectors they call Obama
Hub where we offer over 80 different
data connectors or actually I think it's
over an idea at this point
um that allow you to connect all these

different data formats apis uh like workplace apps structured databases uh to your system uh and you can just ingest this data and then the next steps like Steps two and three are basically just like indexing storing this data and feeding it to the language file uh and so it's actually pretty powerful like we support being able to address like PDF files PowerPoint files Excel files like image files like websites we have like five or six different live streamers at this point and a lot of this is Community Driven too and I think one of the first things people realize with language models was the fact that you can pretty much feed it like any data as long as it was text and it could like try to understand it and then there's as a result there was like a ton of classifications involved on Hub

Machine Learning

absolutely I think that's that's huge both that and what you mentioned earlier about the difference between fine-tuning and training and the context retrieval the fact that you know it doesn't require as much experience to uh like formal ml experience to be able to do some of this contextual um embedding and then also being able to not have to format the data and you know know how to do all the machine learning for that is really allowing so many more people to use these tools and I think that's a huge part of why we're seeing an explosion in this area that now

um yeah machine learning is becoming a lot more accessible to to people in Enterprises and in front-end development in lots of different areas
100 and I think it's interesting because it covers like a spectrum of like um kind of technical expertise uh and what you're seeing these days is uh if you're kind of like a user that has less AI experience uh using a tool like llama index you can get up and running you know uh building a chatbot over your data in three lines of code right address the data uh index that data and query that data and the reason you can actually do that is because of again the fact that language models have radically simplified this like data ETL stack you just like just dump in the data you know index it in some white format and then you can just start asking questions over it
um but I I I I think the the key is also there is like an entire spectrum of like talking full complexity that more kind of like uh sophisticated AI experts are also trying to solve and I think the key is also making sure that you know our toolkit captures the functionality and the extensibility and customizability necessary to so I'll allow them to kind of like play around with these pieces as needed definitely and on that point I want to ask a clarifying question
um that I think people have which is can you clarify a little bit those that's great I didn't know it was only three lines that's amazing so you have the

data connector the indexer in the query could you elaborate a little bit on what an indexer is and kind of more of the process of feeding the data into the llm or giving it to them
yeah totally I think I think when you um when it's like three lines of code it does feel a little bit like magic uh but uh the the all these steps you can customize yourself to but um basically as you said there's the data connectors where you load in the data from say a PDF file or tax file and you just you know or anything that we just talked about like PowerPoints you know databases you load it into a document object and this is a very light document object it's just like a thing that stores a bunch of text that maybe some metadata then you want to Index this document and and the like what indexing basically means is
um there's a
variety of index structures that we provide the most basic one is you uh basically clunk this text up and then you add in a batting to each trunk right and so you um chunk this text up using some sort of a tax flowing strategy and then you add some sort of embedding which is basically a vector of numbers to each chunk that symmetically represents what that tax trunk is about uh now you uh have basically stored this right uh in some sort of after store and you know we offer Integrations with different types of back restore providers like pine cone we did promo

like novas Etc and then once you actually have this then uh you can query this data right use this data stored in fact or sort and the way the query works is you ask a question or you give some tasks to solve and then we would first retrieve the rather than pieces of data from this index using uh like if because if they're stored with embeddings you use like top K and batting similarity lookup and then you retrieve the most relevant text from your Corpus and then you you feed these texts to the language model to synthesize The Final Answer

Retrieval

awesome okay I have two follow-up questions on that I think retrieval is becoming a really important concept um could you talk a little bit more about what exactly retrieval is and how it's relevant to this process
yeah so let me tell you a little bit more about this uh system of kind of like retrieval augmented generation um and so it's a pretty fancy term um but basically the high level is you have a language model uh and then you have this other model called a recruitment model and then you have your data purpose when you have an input request coming in from the user like the input prompt you would first retrieve the relevant pieces of text from your knowledge Corpus using this retrieval model uh and you know it could be a variety of formats but the high level is that it takes in some query and it gives

you back a set of relevant virtual texts
and then you couple that with some
sort of synthesis model or generation
model once you have the retrieve text
you feed it to this generation model and
this generation model will you know
synthesize the final answer for you so
the generational model uh is basically
the language model uh you know the
language model is just saying that can
you know taken about text and then like
synthesize an answer for you now the
retrieval model could be one of a
variety of different things
um these days a lot of people are using
Vector stores to basically store and
index their data so the retrieval model
what it does is it just does like
embedding basis similarity lock up over
your data over the macro store like it
was taking the input query again
inviting for the query uh use that
embedding to look up the talk hit talk K
most similar documents from your
macrosort and then use that as your set
of recruit texts
um it's interesting I think these uh
there's becoming like more and more uh
different types of like retrieval too uh
for instance like uh if you want like a
keyword lookup or like a knowledge graph
look up you could use kind of like
um like relationships between entities
within an olive graph or in between like
different nodes to also do retrieval too
so you go like fetch like not just you
know the current node but also like
relationships between those two

um we have a variety of these types of index structures within lava index that allow for different types of retrieval another very interesting thing is that you know like a retrieval Is Not A New Concept like this thing has been around for a while like if you just take retrieval on its own and uh you know strip away the language model it basically just gets into kind of like how do you best do like information approval and recommendation systems and so there's a lot of Concepts that you can add here too like uh maybe a first safe Passover freewolf that's like uh you first step backs on the natural bucket of items and then you filter it out through some second state class like some rewriting or filtering stuff uh that could be its own recruitment system and so retrieval itself is actually quite interesting there's like you know a decade or a multiple decades research going on there but uh it's also interesting seeing the interaction between the retrieval model with the calibration model because the entire system is a thing that it gives you back the final answer that you would want oh

Structured Data

that's very interesting uh and you actually kind of answered part of my second question which was getting into what if people that are working in Enterprises already have structured data rather than instruct unstructured data and my question was going to be can you

do something maybe on the indexing and chunking level where you're organizing how the information goes in to modify you know the relationships based on the structured data but then you kind of mentioned that you could also use the retrieval step to create some sort of structure because that's something I've run into with my clients is that sometimes there's a spectrum sometimes uh you just want to you know query a long document and see what you can get because you haven't read 10 000 pages of you know
um compliance documents and being able to query them is a great way to to do the initial research but then sometimes you know Enterprises have significant uh infrastructure already in place data infrastructure that has specific structure so yeah what do you think about people that are already have that data and maybe want to build a more like you said sophisticated system
yeah so the part I actually probably didn't talk about was you know the referral happens from something and that something is like a database so like an unstructured uh like uh like different there's different types of databases right there's like unstructured uh databases like key value stores and then like macro stores those will fetch like unstructured documents uh if you will uh and then there's like structured databases like stuff that you could query with SQL
um so those kind of inherently uh like

necessitate like different forms of retrieval so actually probably like the uh for if you already have a bunch of data in like a structure table and you want to query that with natural language uh what you would what you can do is you can uh run like taxes equal on the structured database so for instance um you have this input request coming in from the user you have the structure database you can take this natural language input and you know you wouldn't do it in database a retrieval you would you can actually prompt the language model that output a SQL statement that that you can then execute against uh that structure database and then you get back the set of results that you want so I think that the retrieval move kind of inherently depends on the interface that your data exposes so if it's a structured database it would probably expose some sort of like SQL query interface and then if it's like a vector store for instance it would expose like a top K embedding lookup uh interface because that's more for retrieving like unstructured documents uh interesting

LlamaHubai

that that makes a lot of sense okay thank you for explaining that so you mentioned earlier and I've seen this website before and I think it'll be really useful for people to see you were talking about you have 90 um data connections and that that website's just awesome to look at you

just see the you know you can see all the cards and just tell how much amazing functionality is could you just mention what that web page is and I'll put it in the description so people can check that out
oh yeah for sure uh it's called llamahub.ai
um and uh the website was actually designed by a contributor so I want to give them a little better credit to his name is Jesse uh but yeah it looks nice there's 90 over uh there's over 90 data connectors on there and if you're interested we're always open to a contribution State awesome yeah it looks great good Jesse Jesse crushed it
um so you so there's the data connectors on there you also mentioned that you have multiple uh retriever retrieval mechanisms is that is that on the
um index documentation where we would find the different retrieval methods
yeah so we have an entire like kind of a pretty principal architecture detailed around this idea of like retrieval augmented generation uh there there's a few guides on there in the lava index documentation uh you could Google a lot of index stocks or uh you know that's basically our main website at the moment although we actually have a landing page uh and so uh we have a few guys on on how that works but the idea is that um again you once you actually load in this data you want to index and store this data using some uh some some sort of like uh database or document sort and

then once you store this data you can do retrieval over this data and then synthesis over the recruit data to again build this like overall query interface uh to give you back to the result that you would want
um we have like different reproval methods corresponding to the different index structures that you want to build we also have additional recruitable methods corresponding to some of them are advanced stuff that I described for instance like being able to do brief ranking or being able to kind of like use that lands a you know like a route query to different types of uh like uh kind of like sub data structures or like like
kind of like you have like a variety of uh different indexes and how do you like route a top level query to the different indexes and so all that stuff's in the documentation uh I would definitely check it out it I think hopefully the idea is that to again to get started you can do something super easily and then if you want more advanced functionality uh you can uh the docs should be away so that's all yeah I think yeah what you

How to get started

just said going back to kind of the operating system model being able to dynamically switch between different retrieval mechanisms or different data connectors or different embeddings to like change the personality or memory of the bot I think would be really

interesting and you know something
um yeah seeing these kind of
full-fledged systems where you can
choose maybe what model to use and all
the different parts can be uh changed
depending on the task is really
interesting uh so okay going going back
to a little more basic on how people can
get started so llama index is primarily
python library right now right so can
you talk a little bit more about what
environment let's say someone wants to
get started they want to use the three
lines of code uh basic version one to
start connecting their data to llms can
you talk about you know where is this
how do they they start coding this is it
going to be in their command line is it
going to be in a web brow browser what's
the best way to get started environment
and yeah coding-wise
that's a that's a really good question
uh so llama index is a python package
right now it's publicly available on
pipelines so you can pick install llama
index
um at least in the quick start tutorial
and and a lot of the examples
um it's basically you it's showing you
how to write like a simple python script
either by itself or a notebook that you
can just run and so if you run it in
like a Jupiter notebook uh you know you
could just run each cell and you'll be
able to visualize the results in a
notebook otherwise you could write it in
a script and run it in the command line
so that's just a very Basics that said

we actually do have uh tutorials for how to integrate this as part of some uh you know full sack a web application too and so if you look through the tutorials there are uh are a few of them that show you how to build a full stack app web application we also have like an adjacent repo made by a contributor called like the lava index starter pack which
um shows you how to actually package you know this this thing like your application which is in Python as like a Docker container or as part of like a flat stack end to see bigquery you know from some sort of client front end uh and build like a full-fledged application from awesome and as the previous one um not the starter pack but the other web application guides is that recommended in flask as well
um
I think it's in flask uh let me let me actually double track but I'm like 80.
awesome that sounds great
um yeah I think that is I we like you said we could talk about this data formatting and everything for an hour maybe we could do a follow-up I know we we've discussed doing a follow-up maybe helping people start coding with this a little bit more
um but thank you so much for for answering all these questions and helping people get into this and building this amazing tool and congratulations too on its growth into GitHub it's amazing that you know this

is becoming such an important topic for people and I think llama index is one of the best tools right now to help people get started with that so congratulations on on building this and before we kind of wrap up is there anything um else that you would like to go over or where people can find you and and get started with vlama index or things that you're working on and looking forward to uh no I mean yeah I'd really enjoy being here and if you want to reach out uh there's a lot of links in the documentation to the Twitter the Discord we have a active Discord community so if you play around with Mom index and you ask any questions please feel free to reach out on one of the channels and there's just a lot of discussion going on in it awesome sounds great well thank you so much and hope to have you back soon and I'll keep in touch and looking forward to llama index growing so thank you again thank you awesome