Let's talk about RAG versus fine tuning.
Now, they're both powerful ways
to enhance the capabilities of large language models.
But today you're going to learn about their strengths,
their use cases and how you can choose between them.
So one of the biggest issues with dealing with generative AI right now
is one, enhancing the models,
but also two, dealing with their limitations.
For example, I just recently asked my favorite LLM a simple question.
Who won the Euro 2024 World Championship?
And while this might seem like a simple query for my model,
Well, there's a slight issue.
Because the model wasn't trained on that specific information,
it can't give me an accurate or up to date answer.
At the same time, these popular models are very generalistic.
And so how do we think about specializing them
for specific use cases and adapt them in enterprise applications?
Because your data is one of the most important things that you can work with.
And in the field of AI, using techniques such as RAG or fine tuning
will allow you to supercharge the capabilities that your application delivers.
So in the next few minutes, we're going to learn about both of these techniques,
the differences between them
and where you can start seeing and using them in.
Let's get started.

So let's begin with Retrieval Augmented Generation,
which is a way to increase the capabilities of a model
through retrieving external and up to date information,
augmenting the original prompt that was given to the model,
and then generating a response back
using that context and information.
And this is really powerful because
if we think back about that example
of with the Eurocup,
while the model didn't have the information and context to provide an answer.
And this is one of the big limitations of LLM's,
but this is mitigated in a way with a RAG,
because now instead of having an incorrect
or possibly hallucinated answer,
we're able to work with what's known as a corpus of information.
So this could be data,
this could be PDFs, documents, spreadsheets,
things that are relevant to our specific organization or knowledge
that we need to specialize in.
So when the query comes in this time,
we're working with what's known as a retriever
that's able to pull the correct documents and relative context

to what the question is and then pass that knowledge,
as well as the original prompt to a large language model.
And with its intuition and pre-trained data,
it's able to give us a response back
based on that contextualized information,
which is really, really powerful.
Because we can start to see that
we can get better responses back from a model with our
proprietary and confidential information
without needing to do any retraining on the model.
And this is a great and popular way to enhance the capabilities of a model,
without having to do any fine tuning.
So, as the name implies,
what this involves is taking a large language foundational model.
But this time we're going to be
specializing it in a certain domain or area.
So we're working with labeled and targeted data
that's going to be provided to the model.
And when we do some processing,
we'll have a specialized model for a specific use case
to talk in a certain style,
to have a certain tone that could represent our organization or company.
And so then when a model is queried from a user
or any other type of way,
we'll have a response that gives the correct tone
and output or specialty and a domain that we'd like to receive.
And this is really important because
what we're doing is essentially baking in this context
and intuition into the model.
And it's really important because
this is now a part of the model's
weights versus being supplemented
on top with a technique like rag.

Okay, so we understand how both of these techniques
can enhance a model's accuracy, output and performance.
But let's take a look at their strengths and weaknesses
and some common use cases,
because of the direction that you go in,
can greatly affect a model's performance, its accuracy,
outputs, compute cost, and much, much more.
So let's begin with Retrieval Augmented Generation.
And something that I want to point out here
is that because we're working with a corpus of information and data,
this is perfect for a dynamic data sources such as databases,
and other data repositories
where we want to continuously pull information
and have that up to date for the model to use and understand.

And at the same time,
because we're working with this retriever system
and passing in the information as context in the prompt,
well, that really helps with hallucinations.
And providing the sources for this information is really important
in systems where we need trust and transparency when we're using AI.
So this is fantastic, but
let's also think about this whole system because,
having this efficient retrieval system,
is really important in how we select and pick the data
that we want to provide in that limited context window.
And so maintaining this is also something that you need to think about.
And at the same time,
what we're doing here in this system is effectively
supplementing that information on top of the model.
So we're not essentially enhancing the base model itself,
we're just giving it the relative and contextual information it needs.
Versus fine tuning is a little bit different,
because we're actually baking in
that context and intuition into the model,
while we have greater influence
in essentially how the model behaves and reacts in different situations.
Is it an insurance adjuster?
Can it summarize documents?
Whatever we want the model to do
we can essentially use fine tuning in order to help with that process.
And at the same time, because that is baked into the model's weights itself,
well, that's really great for speed and inference cost
and a variety of other factors that come to running models.
So, for example, we can use smaller prompt context windows
in order to get the responses that we want from the model.
And as we begin to specialize these models,
they can get smaller and smaller for our specific use case.
So it's really great for running these specific, specialized models
in a variety of use cases.
But at the same time we have the same issue of cutoffs.
So up until the point where the model is trained
well, after that, we have no more additional information
that we can give to the model.
So the same issue that we had with the World Cup example.
So both of these have their strengths and weaknesses.
But let's actually see this in some examples and use cases here.

So when you're thinking about choosing between RAG and fine tuning,
it's really important to consider your AI enabled applications,
priorities and requirements.
So namely this starts off with the data.
Is the data that you're working with slow moving or is it fast?

For example, if we need to use,
up to date external information
and have that ready contextually every time we use a model,
then this could be a great use case for RAG.
For example, a product documentation chatbot
where we can continually update the responses with
up to date information.
Now, at the same time, let's think about the industry that you might be in.
Now, fine tuning is really, powerful
for specific industries that have nuances in their writing styles,
terminology, vocabulary.
And so, for example, if we have a legal document Summarizer,
well, this could be a perfect use case for fine tuning.
Now let's think about sources.
This is really important right now
in having transparency behind our models.
And with RAG being able to provide the context
and where the information came from, is really, really great.
And so this could be a great use case again, for that
chat bot for retail insurance and a variety of other specialties where
having that source and information
in the context of the prompt is very important.
But at the same time, we may have things such as past data
in our organization that we can use to train a model.
So let it be accustomed to the data
that we're going to be working with.
For example, again, that legal summarizer
could have past data on different legal cases and documents
that we feed it so that it understands the situation that it's working in
and we have better, more desirable outputs.
So this is cool, but I think the best
situation is a combination of both of these methods.
So let's say we have a financial news reporting service.
Well, we could fine tune it to be native to the industry
of finance and understand all the lingo there.
We could also give it past data of financial records and let it understand
how we work in that specific industry,
but also be able to provide the most up to date sources for news and data
and be able to provide that with a level of confidence and transparency
and trust to the end user who is making that decision
and needs to know the source.
And this is really where a combination of fine tuning and RAG
is so awesome, because we can really build amazing applications
taking advantage of both RAG
as a way to retrieve that information and have it up to date,
but fine tuning to specialize our data,
but also specialize our model in a certain domain.
So, they're both wonderful techniques,

and they have their strengths,
but the choice to use one or combination of both techniques
is up to you and your specific use case and data.
So thank you so much for watching.
As always, if you have any questions about
fine tuning, RAG, or all AI-related topics,
let us know in the comment section below.
Don't forget to like the video
and subscribe to the channel for more content.
Thanks so much for watching!
- Generated with https://kome.ai