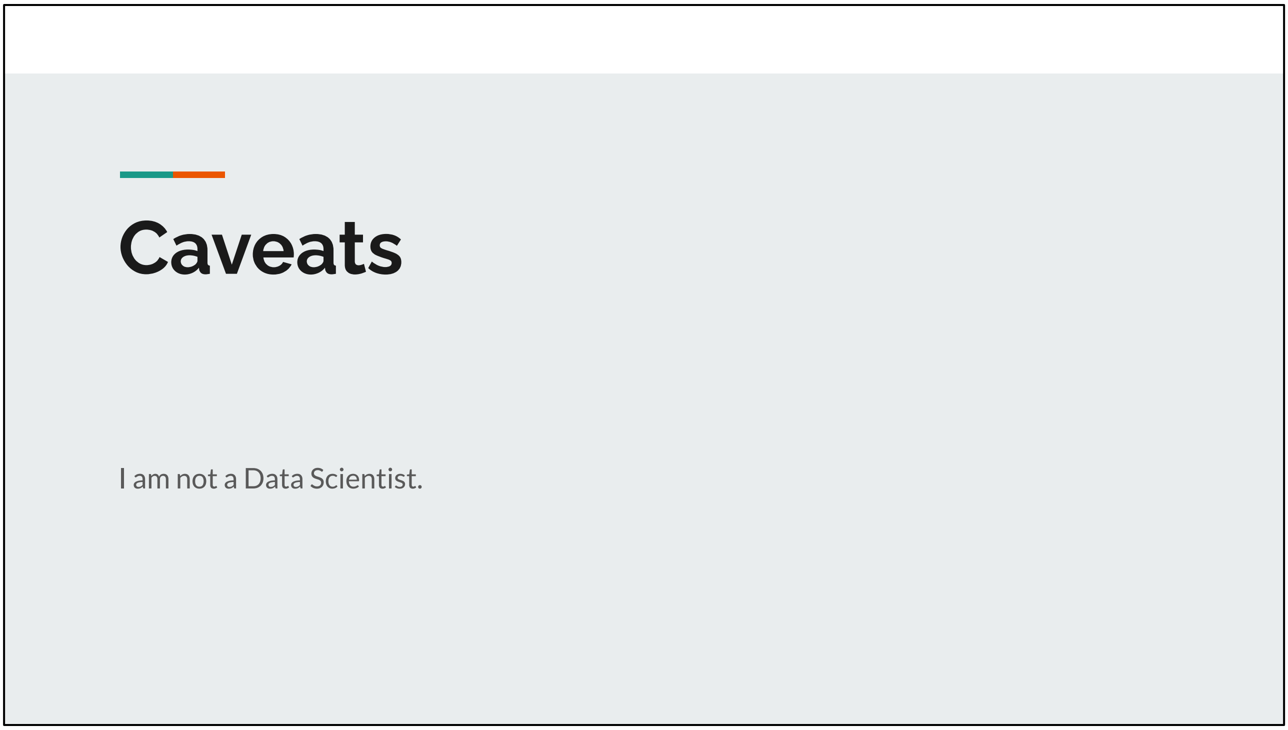
Intro

Hi I’m Tony.

[Christopher Seaman] Tony and I go back to undergraduates, but really reconnected a decade ago when I was looking at out here.
[Christopher Seaman] We've been working in tech startups. You've been working much longer than I have since.
[Anthony Doran] A long time. Yeah. Yeah. We'll say a long time.
[Christopher Seaman] And he currently has his own startup that he sounded called simply put. Is a AI enabled analytics thing, but you'll have more things to say about it.
[Christopher Seaman] And your talk today touches on LMS and everything that was necessary to get up to this point to have LMS and what we're doing and how you do stuff with it.
[Anthony Doran] Yeah. Hopefully we'll have a good.
[Anthony Doran] You have a good picture of the landscape of where we're at right now with LMS and so you can. Figure out how you want to be in part of it, you know, which we hopefully would get that.
[Anthony Doran] Okay.
[Christopher Seaman] Unless there's anything from y'all and questions for again.
[Anthony Doran] Sweet. Cool. Yep. So, like, yeah, I'm done Chris.
[Anthony Doran] I'm Tony, by the way, you know, if you feel like you have a question, just interrupt. It's okay. You know, it would be pretty informal.
[Anthony Doran] But yeah, I'm, I'm a mean Chris met as math people. But I went the route of more of a puzzle solver, just solving puzzles and the land of data.



Caveats

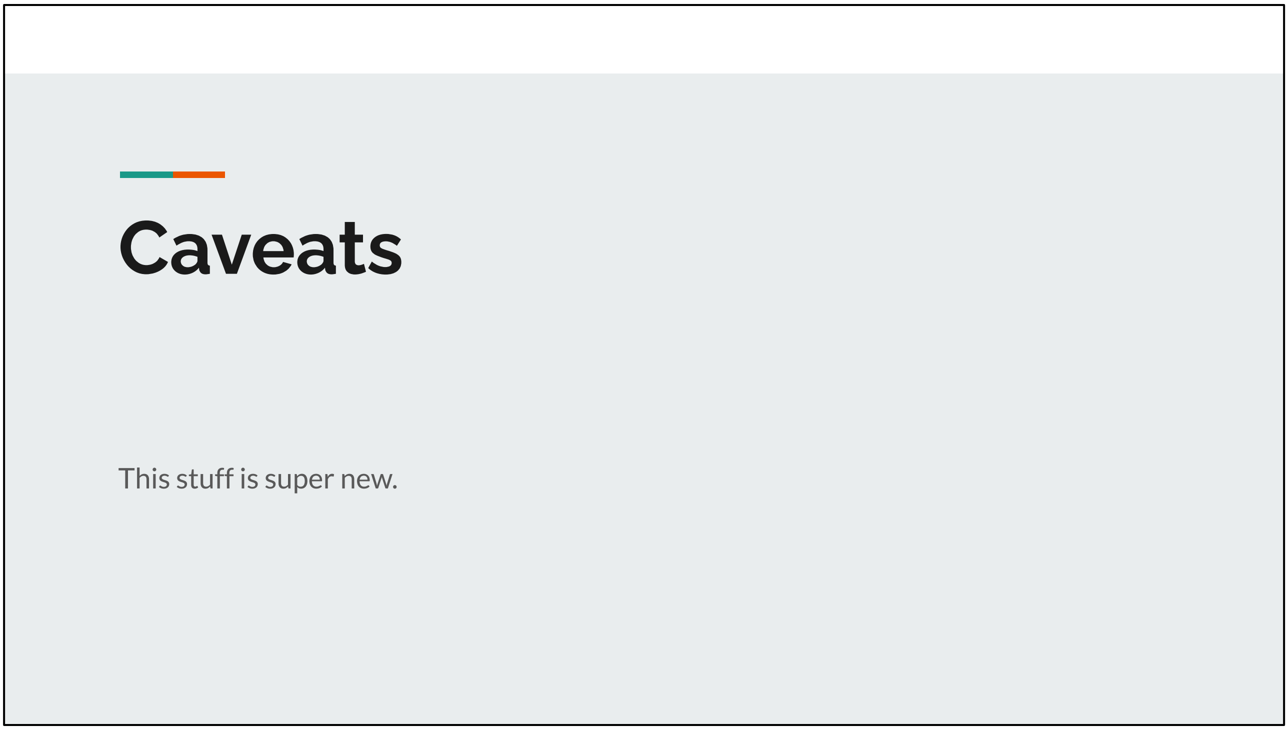
I am not a Data Scientist.

[Anthony Doran] So I've been from the land of like human interaction with data with from visualization to data problems as in like how to pipeline information correctly and I've just been doing that basically for the last forever. We met at Atlassian worked at change.org did some data work over there.

[Anthony Doran] Did some work at Slack. And then in the last year and a half just started, I started my company year and a half ago.

[Anthony Doran] And we'll talk about it a little bit. But first, I'll give you a couple caveats before we start. I'm technically not a data scientist, even though I have a lot of data science experience and I know a lot about it.

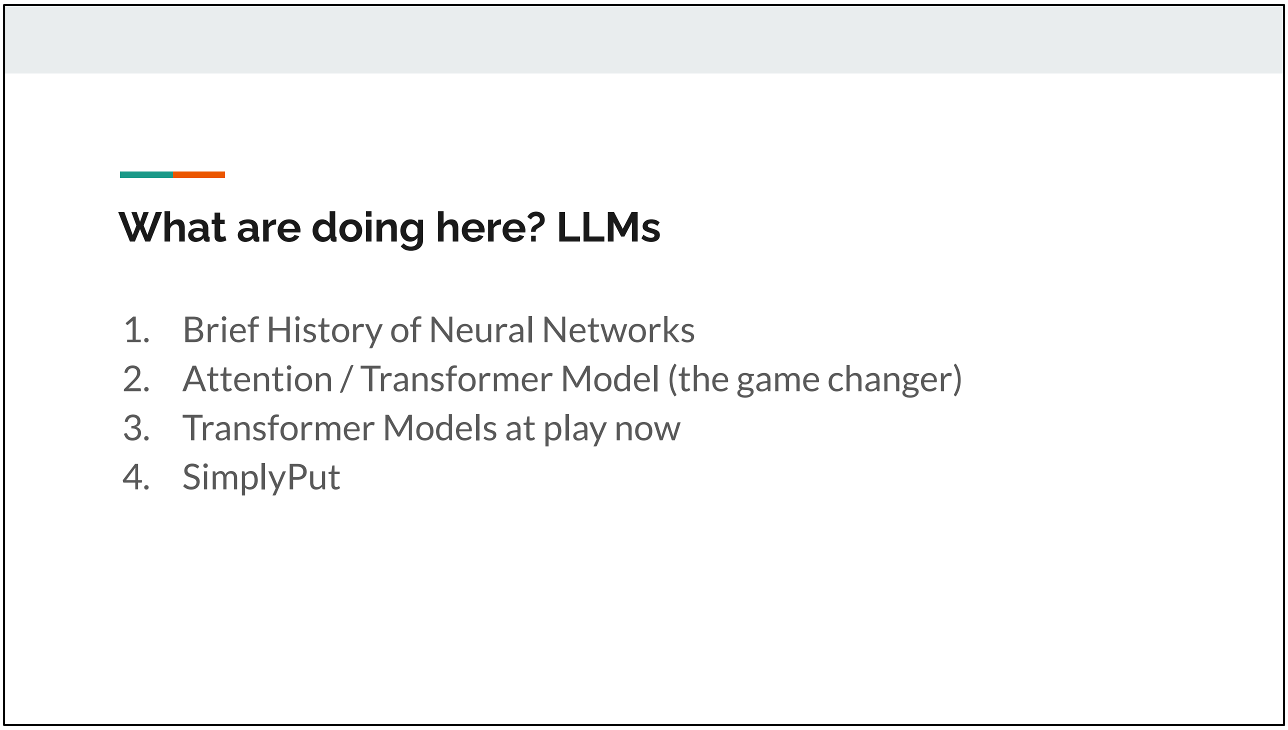
[Anthony Doran] So if we get into the data science type questions, I'm going to be like, maybe, you know, we're going to try and hit this mid level of understanding of where we are, you know, of like a certain. So if there's like a blimp level, and then there's like the raw data level, we're going to sit in this talk somewhere in the middle, maybe a little bit closer to the bottom.



Caveats

This stuff is super new.

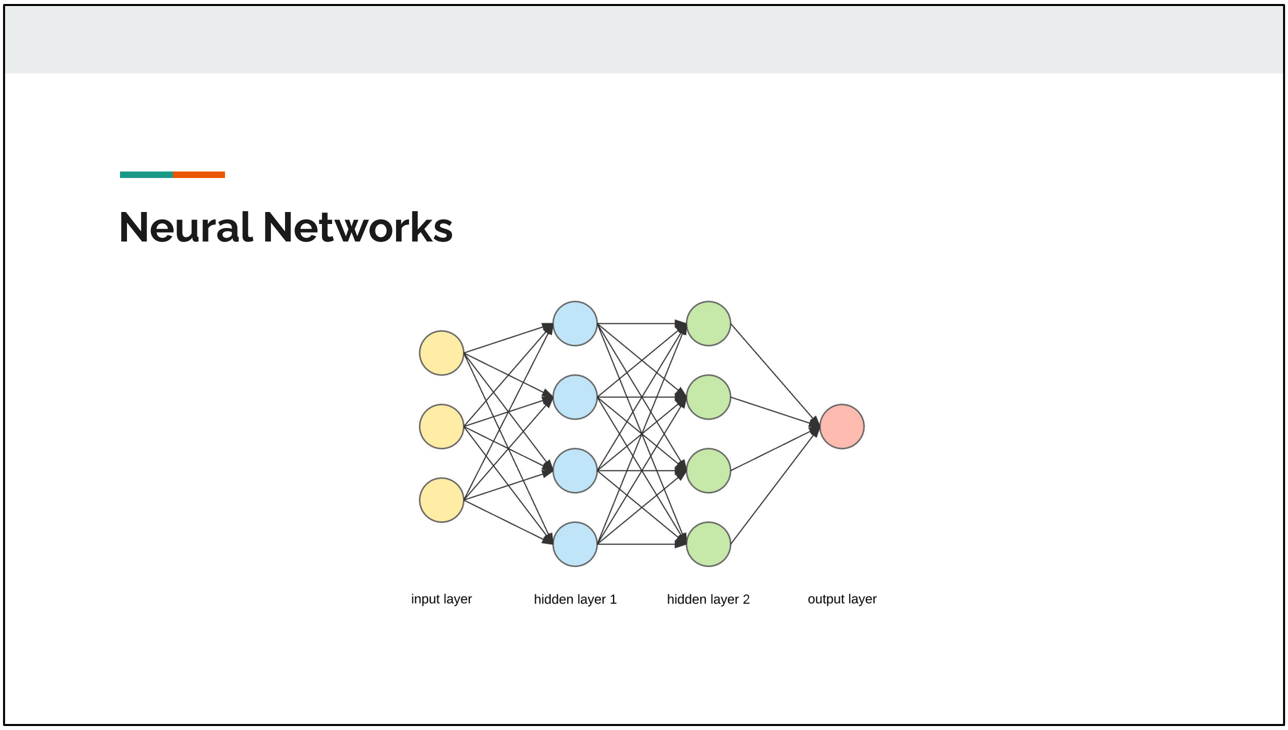
[Anthony Doran] Okay, just want to give that. And then all this stuff is super new that we're talking about.
[Anthony Doran] So it's going to be, it's new to the world, as well as it's new to me. So you might actually know some stuff about this that I don't be very, very curious if this doesn't mesh with anything you're understanding we can talk it up.
[Anthony Doran] Okay. So without.
[Anthony Doran] With that, let's talk about what we're going to do here today. Those caveats.



What are doing here? LLMs

1. Brief History of Neural Networks
2. Attention / Transformer Model (the game changer)
3. Transformer Models at play now
4. SimplyPut

[Anthony Doran] What we're going to talk is about LMS, large language models. Okay. And how we got here is we'll walk through the quick history of the technology will give you appreciation kind of like of how these things work a little bit under the hood.
[Anthony Doran] We're going to talk about what the transformer model is so that, you know, you've probably heard of GPT or chat GPT, the T in that is the transformer and we'll got to understand what that T is. Okay.
[Anthony Doran] And we'll understand the landscape of what transformers are at play that you can interact with big models and play with. And then you'll see how my company interacts with those as well. And then we can explore that.
[Anthony Doran] All good. Okay, great. Oh yeah, on that on that notion talk is a bunch of links. I'm going to refer to those links during this talk, feel free to click on them and explore and see with them.
[Anthony Doran] It's actually going on. Okay.



Neural Networks

[Anthony Doran] So let's go for it. We start with neural networks. I think you guys have probably talked about neural networks before. Right.

[Anthony Doran] Okay, so just as a brief refresher of what these are, this is kind of like our base understanding of how all these things work is that we have an input layer, or some sort of encoded information that we're going to put in. And then we have a series of hidden layers.

[Anthony Doran] And each of those hidden layers are little functions that are going to take parts of that input layer and generate weights for each of those. We go through a series of hidden layers, it could be really big, it could be really, it could be many of them.

[Anthony Doran] Typically not, we have an output layer, it will give us like what we desired, if that output layer. We run a cost function on that output layer, you see how different it is.

[Anthony Doran] If it's a big cost function, we generate our gradient descent, so that we can figure out the new weights, apply the new weights, do this over and over and over again, until we can get weights that are really good that project their output. This is the general, this, this is still used today, this is, but we're going to see how the architecture of this changes over time.

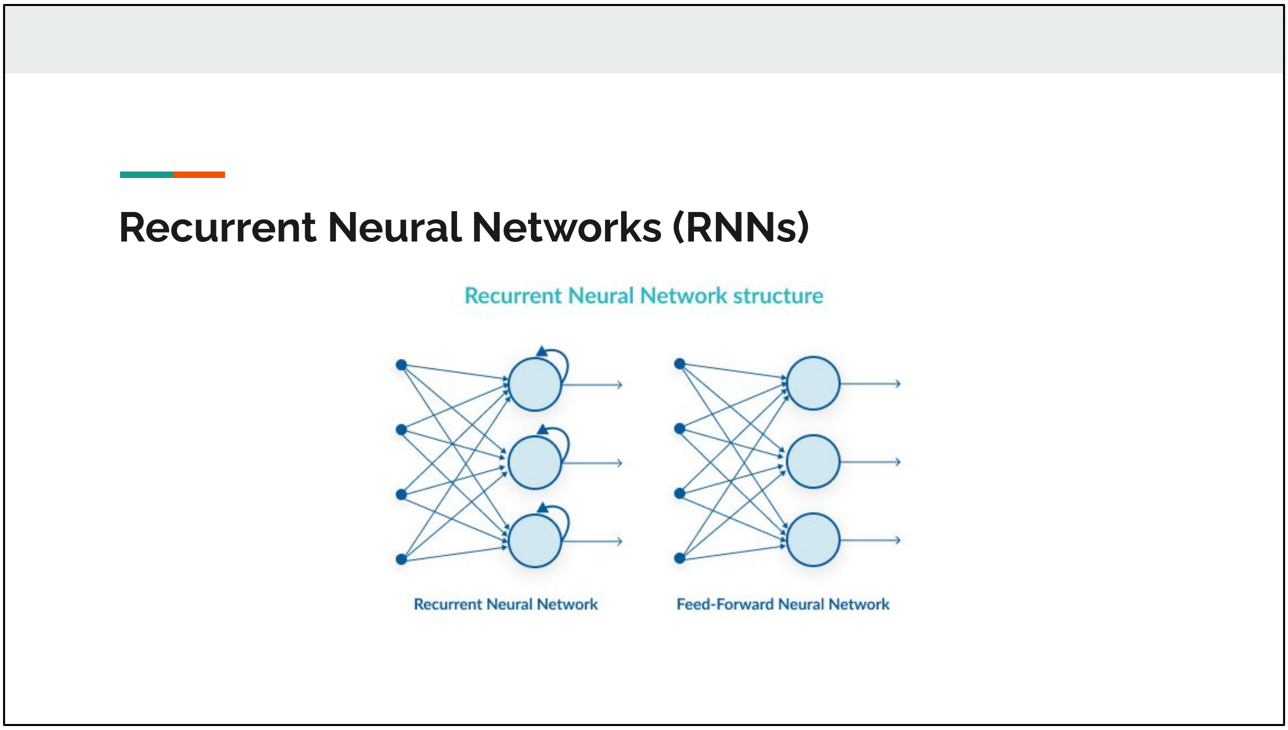
[Anthony Doran] So far so good. Okay, cool. I like the head nods are useful for me. Thank you. All right.

—

Neural Network is at the heart of this. And there are probably better definitions of this. We encode something into a series of numbers or vectors, then we have hidden layers of different sizes that take each vector does a weighted transform on it, for each of them. And the next hidden layer will do the same. Until finally we have an out put vector. We calculate a loss function on that output vector to see how wrong it is, determine the gradient vector to so we can update all the weights. And run it again, see how wrong we are, update the weights. Until we get all the weights in a place where we have minimized our loss function. To me its a giant series of matrix calculations (and is exactly what your graphics cards are good) at

This is the main mechanism that will be used but with some tweaks.

This was done in 1944 by Warrent Mcullough and Walter pitsts in 1952 (confirm dates)

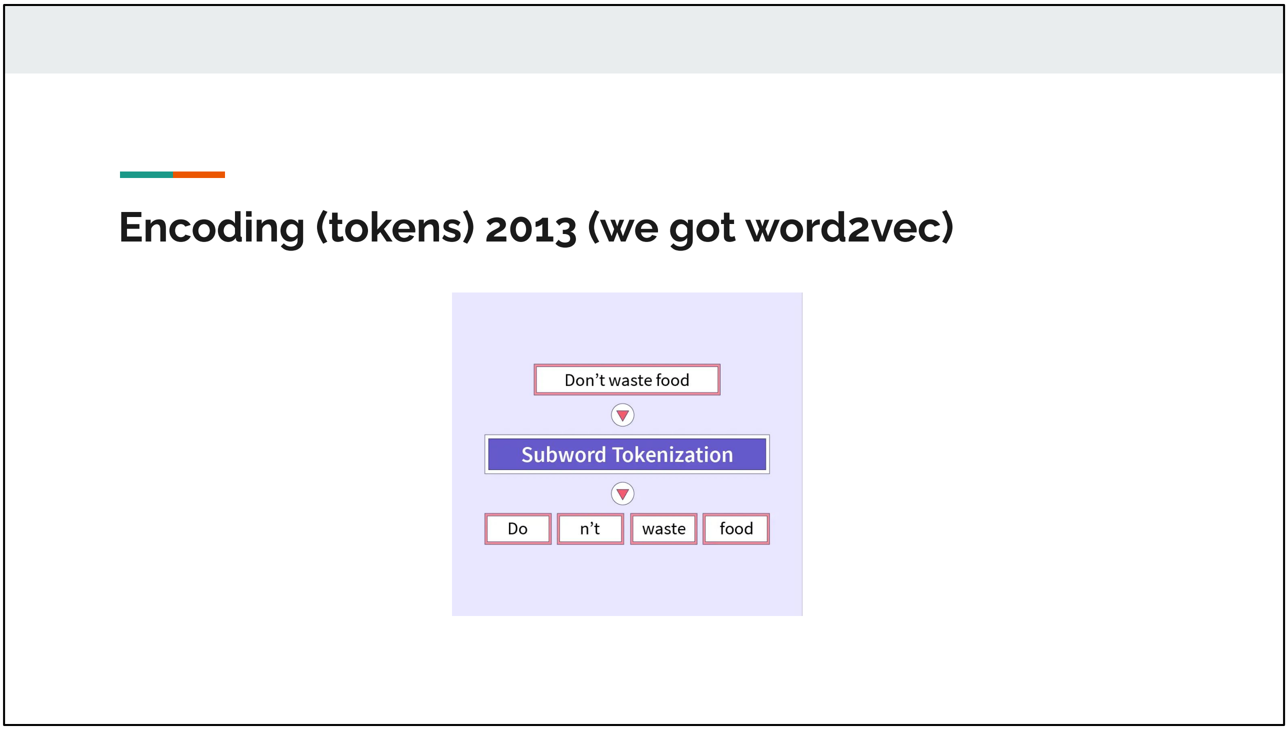


Recurrent Neural Networks (RNNs)

Recurrent Neural Network structure
Recurrent Neural Network
Feed-Forward Neural Network

[Anthony Doran] The innovation on neural networks was comes from RNNs or recurrent neural networks in that there was a needed a better sense of memory in the networks. And so they had have specific nodes that would be able to interact and hold on some sort of memory state as it would process through parts of your input layer.
[Anthony Doran] And so as a result, that became a useful thing for situations which will talk about sequences of information and text is sequence to sequence models will talk about that in a sec. But this, this was, was really used. And,
[Anthony Doran] anyway, this helps with memory, basically. All right. So our history starts all the way back in the ancient year of 2013. Okay. And what happened in 2013 we got this great paper is in the notion dot called the distributed representations of words and phrases and their comp.

—
Recurrent Neural Networks are a version is the a tweak the improves on the base neural network pattern, which allows the network to have a sense of memory, and also allows for the network to deal with variable length input and output. That is needed in language because if i am dealing with a translation from one language to another, we are not always going to know how long the translated string is at the end.



Encoding (tokens) 2013 (we got word2vec)

Don't waste food

Subword Tokenization

Do

n't

waste

food

[Anthony Doran] Compostionality. Oh, God, I can't even pronounce it correctly. Anyway, it was very technical paper in that we have for any language model we do we have to embed the words into numerical formats. And so in this paper we have to say like, Hey, if I've gotten a phrase like don't waste food here and you to split that on to different words and represent that as a giant vector.

[Anthony Doran] So, we have some word tokenization which all the language models use today, but in that paper and specifically in 2000s. Team gave us a whole bunch of efficiencies in language models that wasn't there before.

[Anthony Doran] The, the space was actually much denser than some of the other ones. One of the other things that we got was that their vectors were able to capture much more nuance between the relative definition of one word versus another. Other things that was able to be enabled in this paper was word analogies would have almost like, and almost a mathematical relationship between the different vectors of stuff.

[Anthony Doran] So you can actually almost use like vector addition, or dot, dot products between words and like they would actually be closely analogous. And you also introduced the ability in this, in this model to introduce how phrases could actually be represented similarly to each other as well.

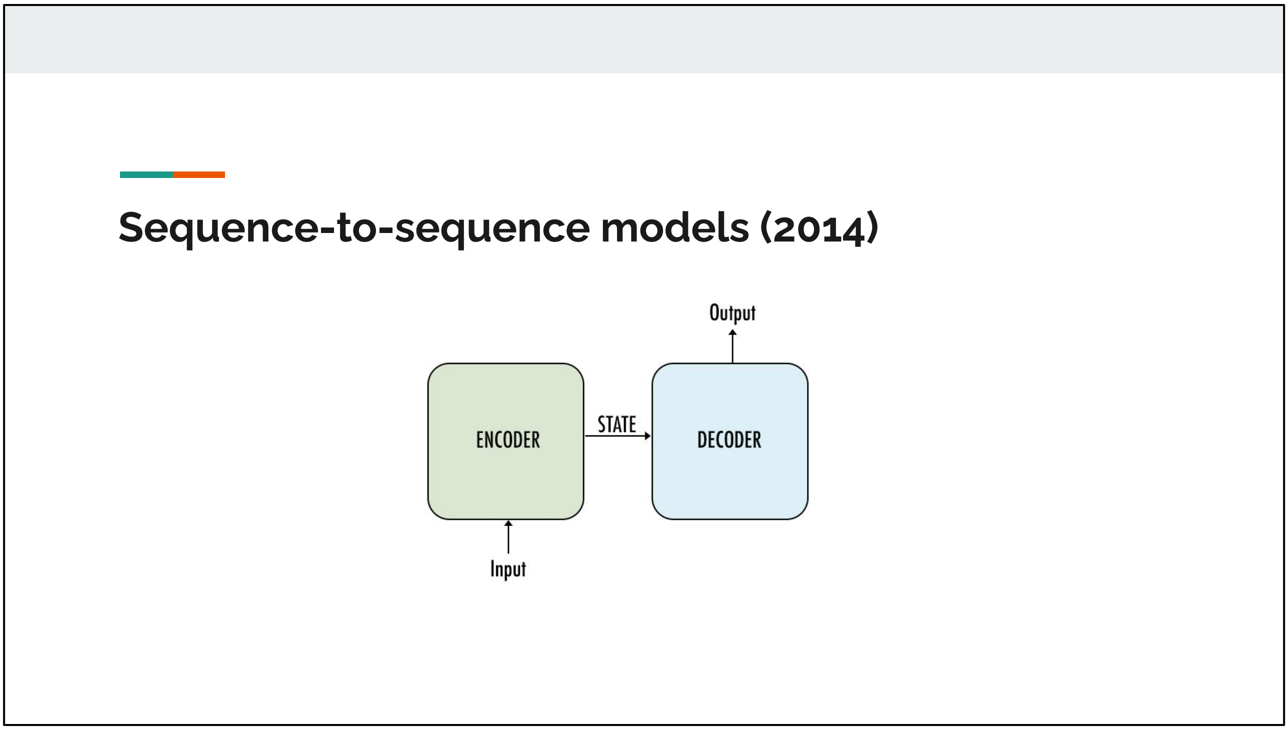
[Anthony Doran] It's called word to VEC. If you see that in the round. Being used, they're referring to this, and this triggered.

[Anthony Doran] This was the basis point for the great stuff that we've been seeing in the last year or so. There are a couple other ones out there, but this is not the other one.

—

-Start our history all the way back to 2013 we get the paper “Distributed Representations of Words and Phrases and their Compositionality”

- Gave us Efficiency,
- Quality Word Vectors (good nuance capture),
- Word Analogies had an unique algebraic representation of this,
- Introduced how to do phrase represnetation….



Sequence-to-sequence models (2014)

Output

ENCODER

STATE

DECODER

Input

[Anthony Doran] Next, we got our sequence to sequence models and the encoder to decoder framework. And we'll see this phrase use all the way up through when we get into transformers.
[Anthony Doran] Just to be clear, when we talk about sequence to sequence models. These are machine learning models that you're taking one sequence and translating it into another sequence.
[Anthony Doran] And a sequence is a sequence of strings and the reason we talking sequence is that the output is of unknown fixed length. Right. So if you're translating from English to French.
[Anthony Doran] And you're taking an English sentence, you don't know the output vector size, or how many, how many points are going to be in the vector size. And so this is why these are called sequence to sequence.
[Anthony Doran] And it's kind of a different modeling, because most of the time when you're doing. Normal network you always know what the size of your output is. Right.
[Anthony Doran] So, for examples. So one language to another language is a sequence to sequence model. One paragraph. So if you have a paragraph and you're generating a summary.
[Anthony Doran] That's a sequence to sequence model conversion. Right. So if you have a current conversation and you're trying to get what the next word and phrase of the conversation is going to be another sequence to sequence one.
[Anthony Doran] And even like sentence correction, like I have a sentence. There's something wrong with the second sentence fix it. Sequence to sequence model, all of these fall into this camp.
[Anthony Doran] And they use this encoded decoder. So, architecture for your neural

network. And what's going on here simply is in the encoder, you'll have your input layer going into that, you'll have your RNN.
[Anthony Doran] In fact, baked in there. And instead of the RNN just going right to the output vector where your probabilities are going to be.
[Anthony Doran] It's going to generate a layer, another layer for the decoder. And so you'll have a target.
[Anthony Doran] So let's say we have a phrase, well, we'll go through this again but let's say you have an input phrase like something in English. You go through, go through here and you know it's supposed to translate to something in French.
[Anthony Doran] So this is going to go through its own the targets can go through its own RNN. And the input's going to go through its own, but the input's going to generate a hidden, hidden layer for the decoder to then generate the probabilities of how successful it was.
[Anthony Doran] So, that's going to be very high level. And be used in the transformer at that point.
[Anthony Doran] Okay. And that was in 2014 we got back and it started triggering, triggering triggering people to really rethink about how they were doing their RNNs. But there was another problem with this model.
[Anthony Doran] And that state vector that they would share between the encoder and the decoder networks was of a very fixed state vector. It was only up. And it was only one layer where the decoder when it was trying to get the probabilities of the target.

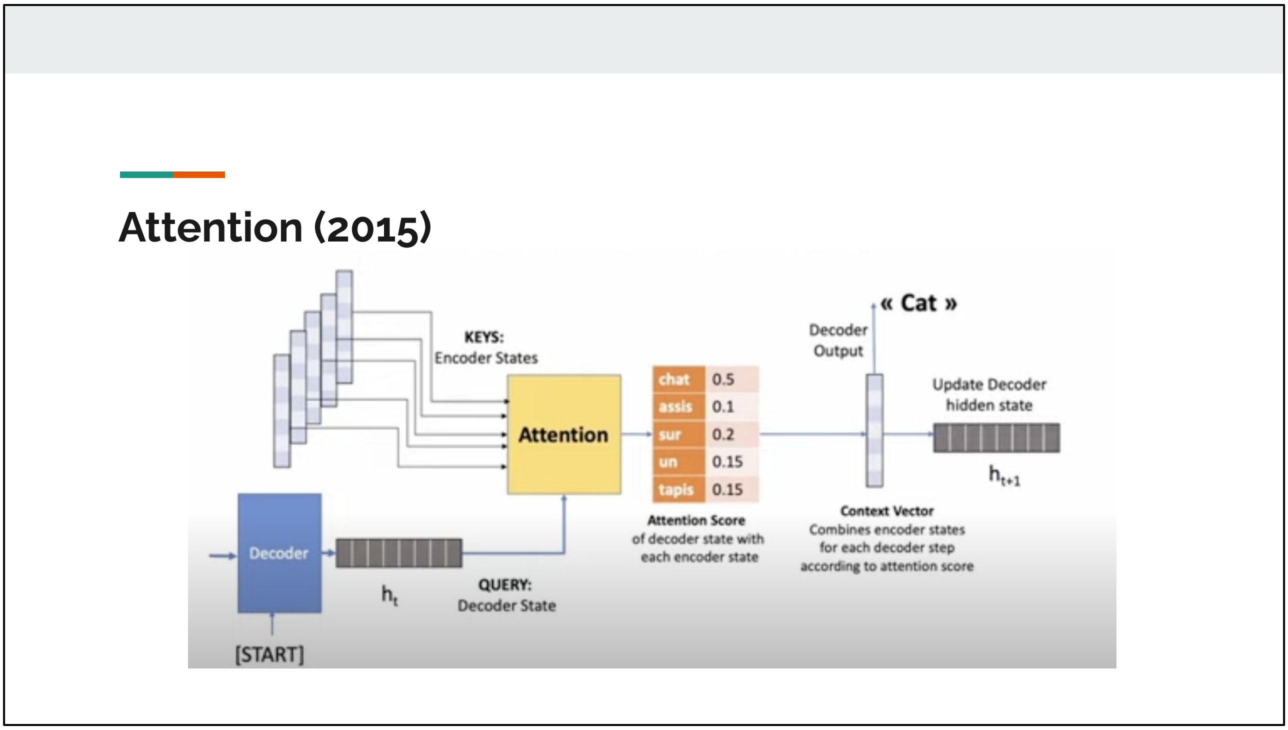
—

In 2014 we got the Encoder Decoder architecture. This introduced us to the terminology of sequence-to-sequence models

1. One phrase from one language to another language
2. One paragraph to a summary
3. A Conversation so far, into the next response from the a conversation
4. An incorrect sentence to a corrected sentence

Each Encoder Decoder has its on RNN. except the encoder role here is to provide a state vector that will become a hidden layer for the decoder. The Decoders role then becomes to translate the encoded properties of the input into the output sequence

This pattern will end up being foundational in the Transformer Model.



Attention (2015)

[Anthony Doran] It needed to know much more nuanced stuff about how the original text was actually being encoded. And so this intention mechanism was
[Anthony Doran] in 2015. And it's complicated.
[Anthony Doran] I'm not going to do it justice. So I have an animation that will help us walk us through. But in essence, what it's going to do is it's going to give basically how much attention each word.
[Anthony Doran] How important each word is to the output. And then the two thing one of the cool things is most machine learning networks. When it works you're totally in a black box situation you don't know why it worked.
[Anthony Doran] Right. With attention scores going into the model you'll be able to get pictures like I was focusing on this word. I was focusing on this word so it unblack box, a bunch of stuff. And then the machine process. Let's, let's go to the animation that I found that does it really good justice.
[Anthony Doran] This is in your in your notes as well. That's the, that's about attention.
[Anthony Doran] Yeah, so. So let's say we have this initial phrase.
[Anthony Doran] And we have our hidden layer and it's generated our initial decoder state that would be passed over to the network. So we get some alignment scores based off the state.
[Anthony Doran] Normalizes them. And then generates its attention weights.
[Anthony Doran] Multiplies through. Then generate a context vector of which it shares with the decoder.
[Anthony Doran] Generates a new state. And generates the first word.
[Anthony Doran] Passes the state back. Goes through the alignment attention weights again.

[Anthony Doran] New context vector. New word.
[Anthony Doran] And it keeps on going until it reaches the end since. So these are quite a recursive process. Right. So,
[Anthony Doran] Anyway, hopefully I'm hitting this mid level. Pretty well, but don't worry. We'll, we'll get, we'll get through this.
[Anthony Doran] Okay. So that's, that's, that became the attention. Part.

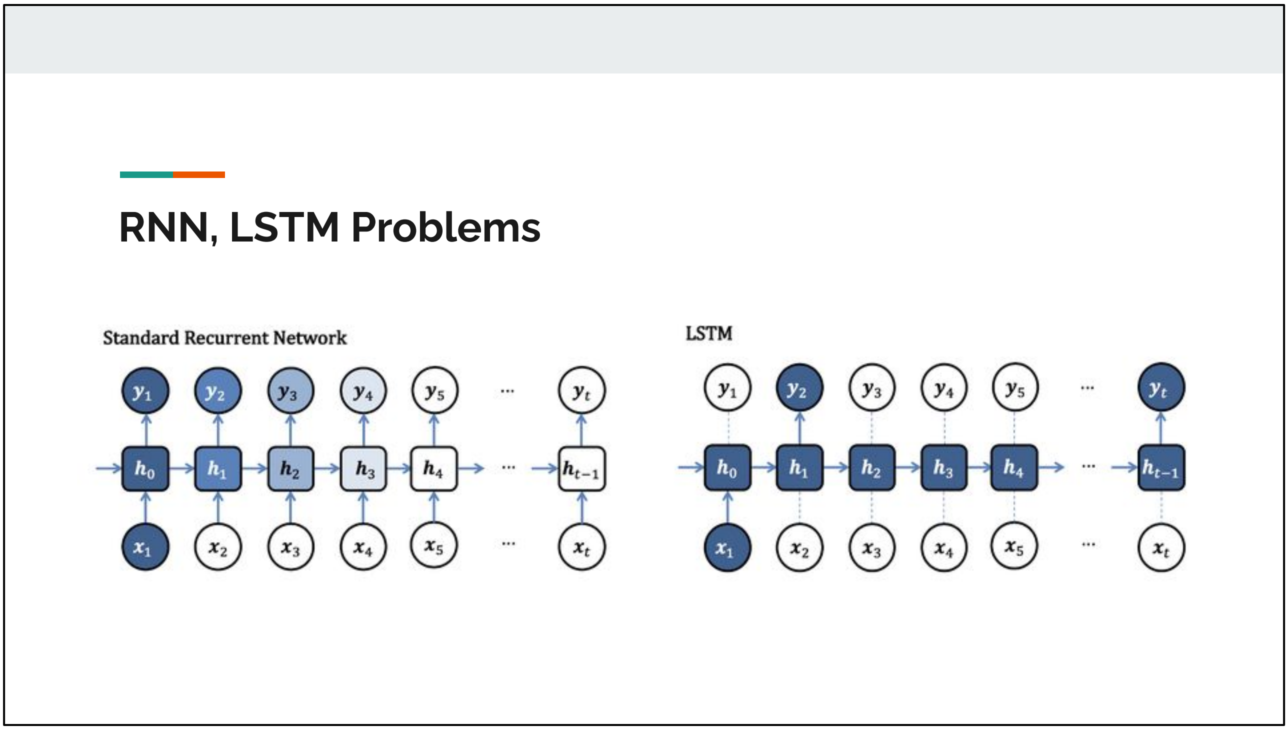
—

There was a problem with the previous approach to the encoder-decoder was that the state vector was just one fixed length vector.. This introduced the Attention. Instead of being dependent on that one state vector.

Decoder can focus on specific tokens on the input sequence. At each step the decoder can form new context vectors by combining new hidden states from the input sequence.

This allows it to handle long Sequences
It gives us the ability to see what is thinking, through seeing which words it is currently thinking are important.

https://erdem.pl/2021/05/introduction-to-attention-mechanism



RNN, LSTM Problems

Standard Recurrent Network
y_1 y_2 y_3 y_4 y_5 ... y_t
h_0 h_1 h_2 h_3 h_4 ... h_{t-1}
x_1 x_2 x_3 x_4 x_5 ... x_t
LSTM
y_1 y_2 y_3 y_4 y_5 ... y_t
h_0 h_1 h_2 h_3 h_4 ... h_{t-1}
x_1 x_2 x_3 x_4 x_5 ... x_t

[Anthony Doran] But we still had two big problems. One is that in our ends, there's.
[Anthony Doran] Two main things that we needed to deal with. One is that there's thing called vanishing gradient. Problem.
[Anthony Doran] I don't know if you guys ever heard of this. Okay. So in our ends, if you have a memory node that is going to recurse over and over and over, the tendency for the weights is to do. Of the importance of that memory is the tendency is for that weight to either explode in importance or go to zero.
[Anthony Doran] As it keeps on going through. You know, just the, and, and so that, that became a problem.
[Anthony Doran] That becomes a problem for our ends. And LSTMs are also a recursive process when it's building up its weights.
[Anthony Doran] It handles the memory a little bit better, but still this nature of recursion is slowing. Down the ability for all these, all these models to train over big amounts of data.
[Anthony Doran] Because every time you going through and generating probability, you have to rebuild the weights for each word going through. I think that yeah, I think there's techniques for handling it, but yes, I would imagine it would handle that as well.
[Christopher Seaman] To be honest, but yes, overfitting. Yes.

—

So even with the Attention there were still 2 problems

This is called the “Vanishing gradient problem” This occurs that when the back propagation happens the weight to improve the model tend to do either Explode or vanish as the the input sequence gets longer

While LSTM’s handle the vanishing gradient problem better Training was slow because each words output embedding was dependent on each words embedding before it.

Attention! (2017)

arXiv:1706.03762v5 [cs.CL] 6 Dec 2017

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

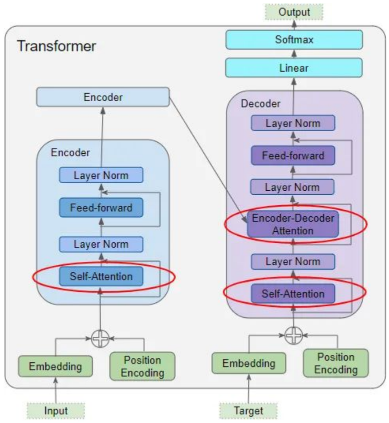
1 Introduction

[Anthony Doran] And then, and then the world changed in 2017. Or at least the world of of of of models change in 2017 with this big career is attention is all you need.
[Anthony Doran] This basically killed all RNNs for machine learning in L in LMS. No more recursive process.
[Anthony Doran] And this paper introduced what is known as the transformer model. And this is, this is a picture of it.
[Anthony Doran] The attention that tension mechanism of getting the relative meaning between each words. Becomes really useful so.

—

Then in 2017 This paper is released. And RNNS & LTSM's were gone.This gave us the transformer

Transformer



[Anthony Doran] And it's used in the encoding layer in the decoding layer and in this middle. When it's able to mix.
[Anthony Doran] So think of it. The best way to think about it is this way in the encoding layer. It's going to generate the relative scores of the importance of the words with every word in the input layer.
[Anthony Doran] So if I have a five word input layer, it's going to generate scores of how important each word is to each other word. For the input phrase.
[Anthony Doran] Goes through hidden layers and multiple encoders to generate the attention keys that is going to pass into the decoder later. Then the target phrase, the phrase where we're trying to match it to.
[Anthony Doran] It's going to go to, hey, how important are each of these words. The only difference is that the decoder layer only has to, has to go from forward.
[Anthony Doran] It can't look forward. So it's going to start with one word.
[Anthony Doran] It's like, oh, yeah, this is the most important word. Then the next word in the target is like, oh, yeah, what's the relative words for these two.
[Anthony Doran] And then the third word. Which of this balances course of the third, so it has to repeat, do it scores that way.
[Anthony Doran] And then when it gets to the encoding there. It does. So this is doing.
[Anthony Doran] Importance of the words for the target. Importance of the relative words for the input.
[Anthony Doran] And then this does the importance of the targets to the. To the input.
[Anthony Doran] And then passes it all the way forward to then get your. Probability.
[Anthony Doran] Okay. I'm going to keep on going because this is just fun.
[Anthony Doran] It's gets crazier too. Okay. We're that.

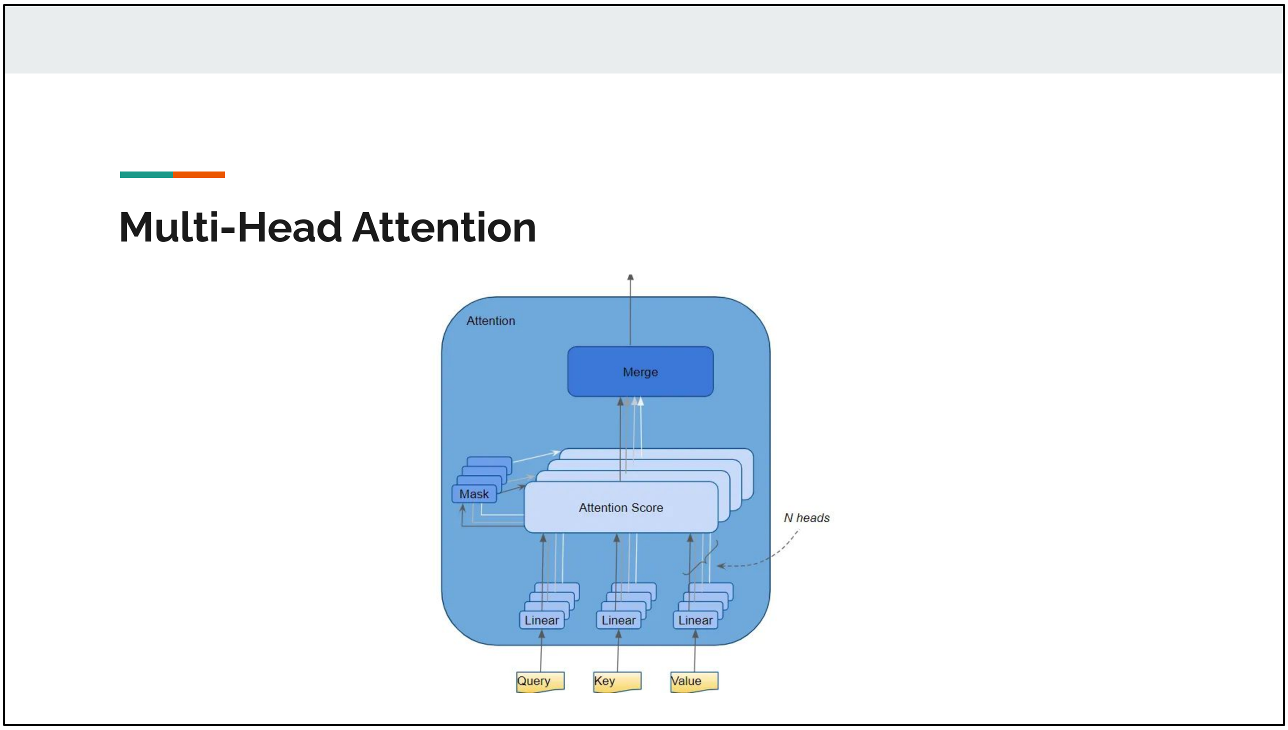
[Christopher Seaman] That goes back to what you're saying before with the sequence. Yeah.
[Christopher Seaman] So that's going to this dense, many dimensional, not human readable.
[Anthony Doran] Yeah.
[Christopher Seaman] Everything is a matrix.
[Anthony Doran] Everything is just matrix transformation. Okay.
[Anthony Doran] And it's wild. And it's, it's a matrix of vectors. It's not even a matrix of numbers.
[Christopher Seaman] The middle of this diagram. And code.
[Christopher Seaman] The box on the top left. To the purple.
[Christopher Seaman] That's all. Vectors.
[Anthony Doran] Yep. It's many vectors and actually.
[Anthony Doran] This is not doing a good job of it. It's many encoders. I think sometimes it's actually six of them.

—

Self Attention is just mechanism to generate weights for each pair of words in the sequence. This is the mechanism that alleviates the need for RNNs because now each word knows is relevance to each other word.

This "Attention" step happens in three places

1. The Encoder (where its connecting the importance to each word in the input sequence to the other words in the input sequence)
2. The Decoder (where its connecting the importance to each word in the target sequence to the other words in the target sequence) This works slightly differently, in that the decoder, because it only works on pairs of words that come before it
3. And the Encoder-Decoder (where its connecting the importance to each word in the input sequence to the other words in the output sequence) this worked just like it did the Encoder-Decoder model with RNNs,

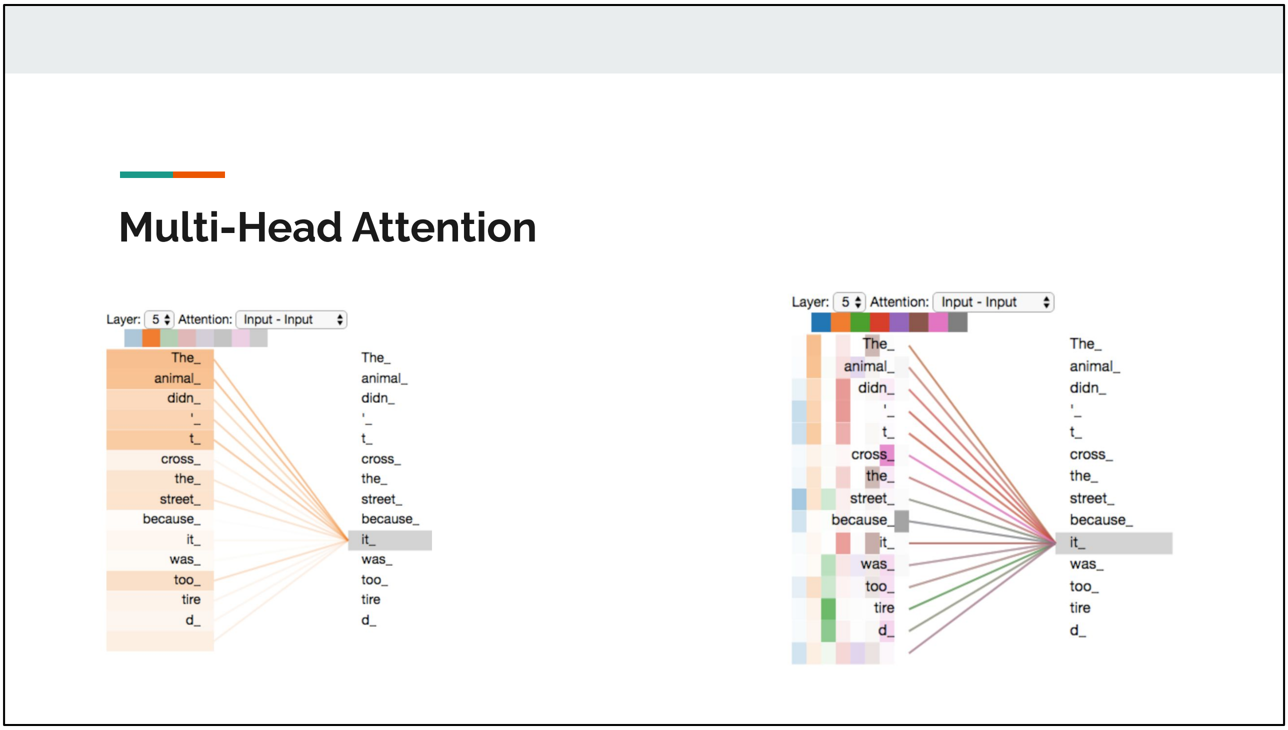


Multi-Head Attention

[Anthony Doran] And for, and this is six of them too. And it's going to keep on doing it for each six as it goes through, which is wild. One of the cool things is.
[Anthony Doran] In this paper also they introduced the concept of multi head attention, which is really useful because. Well, just shows you how much computation is actually going on in here, but it doesn't just do it from one relative meeting between each of the words.
[Anthony Doran] It allows you to have multiple relative meetings for input phrases in there, because there's many different ways of how words can actually be related to each other. And the cool thing is, it doesn't know what they all are.
[Anthony Doran] It's going to pick up on the nuance of why this word should be associated to this other one. And this one should be the one.

—

Now to make things even crazier each attention step actually is running a Multi-Head Attention. (this is all referenced in that paper). For each attention head, its generating the importance the word is to each of the word is, but allowing for different interpretations on what “important” might mean.

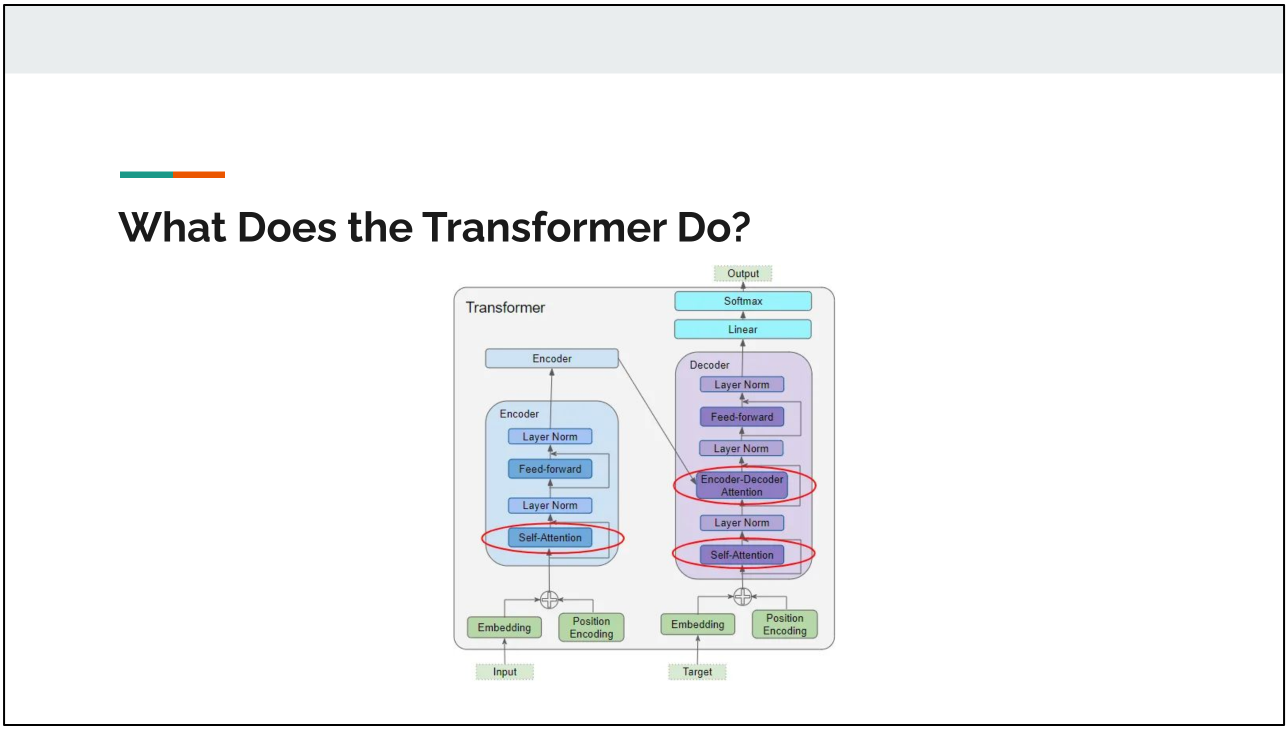


Multi-Head Attention

[Anthony Doran] This is just kind of a nice visual representation of it. So here it's kind of showing on the left with one way of.
[Anthony Doran] Showing the relative scores. And this is what the color coding would be with multiple different ways of. Of the association can be.
[Christopher Seaman] So, when it's passing through your network to actually respond to get to the big bone state, it's actually considering possibly different meanings for quote unquote.
[Anthony Doran] From. I like to think of it as.
[Anthony Doran] How is this word. What's one way to think of these words related to each other one way is like. Grammatical are the same parts of speech.
[Anthony Doran] Another way is like, are they in the same vocabulary level as of expertise. You know, is it the same language.
[Anthony Doran] You know, but we don't know which ones are the important ones. The training.
[Anthony Doran] Mechanism starts figuring out which are the right ones to actually use.
[Christopher Seaman] Or if we had a model that was based on that will start to be in different categories of apologies.
[Anthony Doran] Yeah. Yeah.
[Anthony Doran] So that multi head attention. So.

—

One Attention head deals with one way each word is associated to other words in the sequence. But using multi-head allows the backpropagation to learn different ways one word might be related to each another word. This allows for multiple ways that words might be related to each other in the sequence



What Does the Transformer Do?

[Anthony Doran] So let's go back to the transformer. Now.
[Anthony Doran] We have no recurrent. Processing, right.
[Anthony Doran] Everything is what they call feed forward. Pretty much.
[Anthony Doran] We don't, we just need to go for we don't need to rebuild weights as we go through the actual input sequence, which parallelize makes the model much more parallelizable. Huge efficiency boost.
[Anthony Doran] We also have. Structured the network so that it can not just learn just weird noise of what's going on, but actually structured it so it can actually focus in on the actual meanings and relative meanings between the words and learn subtleties.
[Anthony Doran] And that paper when they were doing it, they didn't realize how important was actually going to be. Which is pretty wild.

—

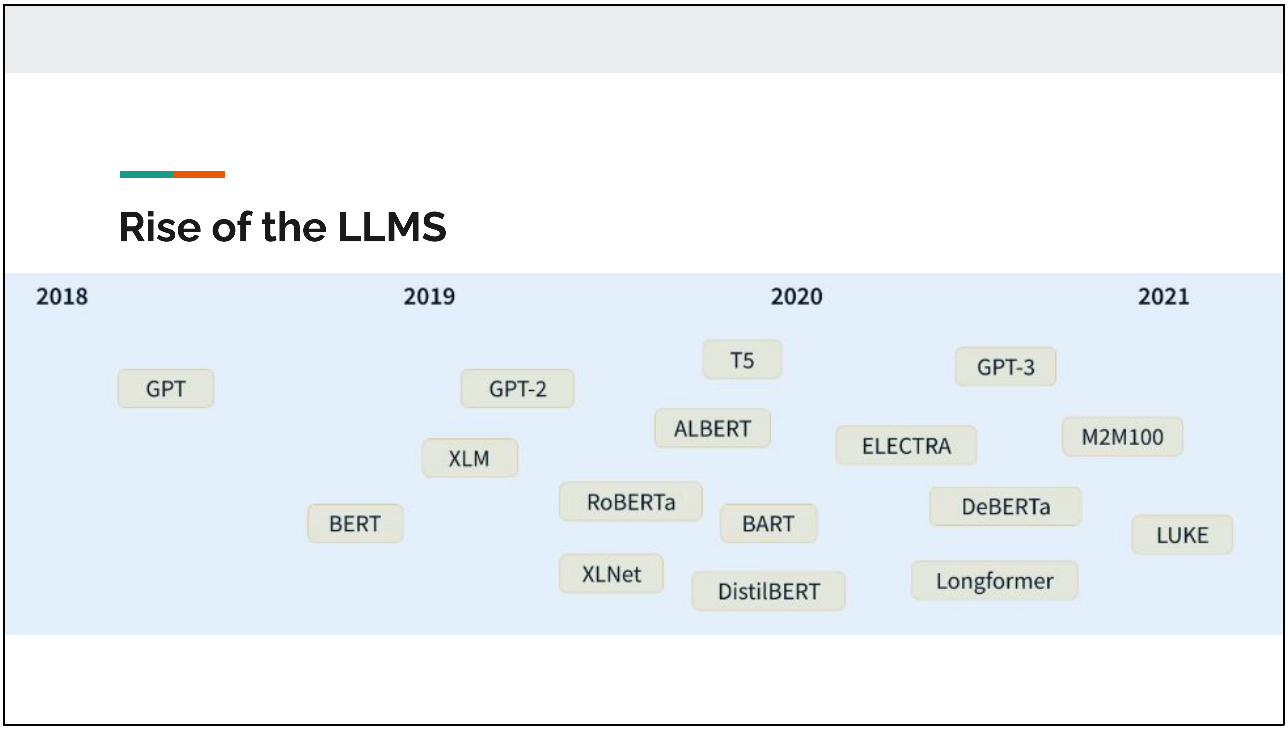
So to recap on the what the Transformer does

We have an input phrase and target phrase.
We embed the phrase into tokens, which are different vectors
We then get attention scores for how each of those token are important to each other
Pass through some hidden vectors and produce attention key vectors for the the decoder.

The decoder has a target phrase
It is imbedded into token vectors

Generates attention scores but only for for the pairs of words that come before it
Then generates attention scores based on the input attention key vectors
Pass through some hidden layers
Produce output probabilities for which word should be next

We do that over many examples
Calculate our losses
Update our all our weights. Then we do it again.
Wild right?



Rise of the LLMS

2018 2019 2020 2021

GPT
BERT
XLM
GPT-2
RoBERTa
XLNet
T5
ALBERT
BART
DistilBERT
ELECTRA
GPT-3
DeBERTa
Longformer
M2M100
LUKE

[Anthony Doran] And so this is the timeline after 2017 is how how models start some of the models that started happening. So the first one large language, and this is the rise of how the large language models came out. So in 2018 we get.
[Anthony Doran] GPT, which is the first one kind of on out there, which had 170 million different parameters, and it was trained on basically the entirety of the web on the Internet. Okay. And the model that the sequence model it was using was next word prediction. So given this word, what's the next word that should show up given this word in this word.
[Anthony Doran] What's the next word. Next was Burt.
[Anthony Doran] There's also a large language model using the transformer, but it's modeling that it used with something called a mask language. Mass language training is like you have an input phrase, and you take one word out.
[Anthony Doran] What's the probability you got that word, put it through the model it puts the right word there. You know, and it does it does it's learning that way.
[Anthony Doran] It does pretty well. Excel net.
[Anthony Doran] I'm just noting some of them. I'm not going to talk through usual Excel. I'm just doing the ones I think we're interesting Excel net. Did something new where it did something called a permutation language modeling.
[Anthony Doran] I tried to read it don't understand. Okay.
[Anthony Doran] But it's linked to there for you, you know, it's, it's, and, but the, the way reason I mentioned it is that it beat the pants off. So, you know, it's not a word.
[Anthony Doran] Burt, Burt got destroyed by Excel net in accuracy. Then we got to knowledge distillation, which the still bird was actually kind of interesting, that these models became so big.

[Anthony Doran] Knowledge distillation is a kind of a process to take one of these large language models. Sacrifice some of the accuracy, but do a radical size compression on it so that it could run actually superficially and super fast.
[Anthony Doran] So that might be actually useful to you at some point, you know, because there's some models that are just so big. It's actually hard.
[Anthony Doran] Next major thing to note is the T five, which I think really blew up the world. A part partially below the world, because it introduced something called text to text modeling.
[Anthony Doran] And text to text modeling allowed the language model to learn what kind of NLP tasks should I be executing now. So, instead, so you can give it instructions like I want you to solve this problem and the model will actually know what kind of problem you're trying to solve.
[Anthony Doran] Which is wild so if it's like, hey, I need you to translate something. Okay, you tell it translate this and it will learn to translate it. If you tell it to what should be the word at the end of the sentence, it'll know to do that.
[Anthony Doran] And so it gave the models to learn how to do multiple NLP sequence to sequence tasks. Okay, then GPT three came in the middle of 2020.
[Anthony Doran] And then was originally 170 million parameters. GPT three was 175 billion parameters. You know, and so they got blue up, you know, during this very short time period.
[Anthony Doran] I'm training this. And this gave birth to chat GPT, which came in 2022.
[Anthony Doran] I forgot the exact date. I'm sorry. Yeah. And I'm chat GPT four, which you need kind of special access to, but I think Bing has Bing searches I think is using GPT four, you can use that there.
[Anthony Doran] And I have developer access we can play with it a little bit at the end here, which is, which is cool. So this started as these models got so big.
[Anthony Doran] So big, so fast. It was basically to kill the RNNs.
[Christopher Seaman] So how do I get to 175,000 parameters, and why didn't that hospital. Um,
[Anthony Doran] Yeah, it's cool. My answer is still the same. You know, if you tried to do 175 billion parameters on the RNN network.
[Anthony Doran] You know, you just wouldn't have enough time. You wouldn't have the computer.
[Christopher Seaman] Yeah.
[Anthony Doran] Yeah. So here I can go and you can start taking advantage of all the GPUs and GPUs are made for basically this kind of calculation because there's no is it's not going like this update wait word update wait word update wait word update wait.
[Christopher Seaman] It's doing all these words, flush it through. So I can either do it after I've already trained this thing and add that to the model doesn't matter the order that I did.
[Christopher Seaman] Or I can just wait for additional computing. Yeah.

—

LLMs are Transformers learned on the Massive size of Data with a massive size of parameters.

GPT was first LLMs 170M parameters, trained on the Data of the Web, using Next Word Prediction

BERT Transfomer
Pre trained on Masked Language model

XLNet
Introduces (Permutation Language modeling) and beat BERT in a bunch of tasks

Knowledge Distillation
Disitillibert Is a model that mimics that output of another model, but tries to cut down the size of the model to get rid of stuff

T5
Introduced Text to Text modelling which Text to Text learning that allowed models to learn a variety of NLP tasks in the same model. This will infer what kind of task am i doing from the input text.

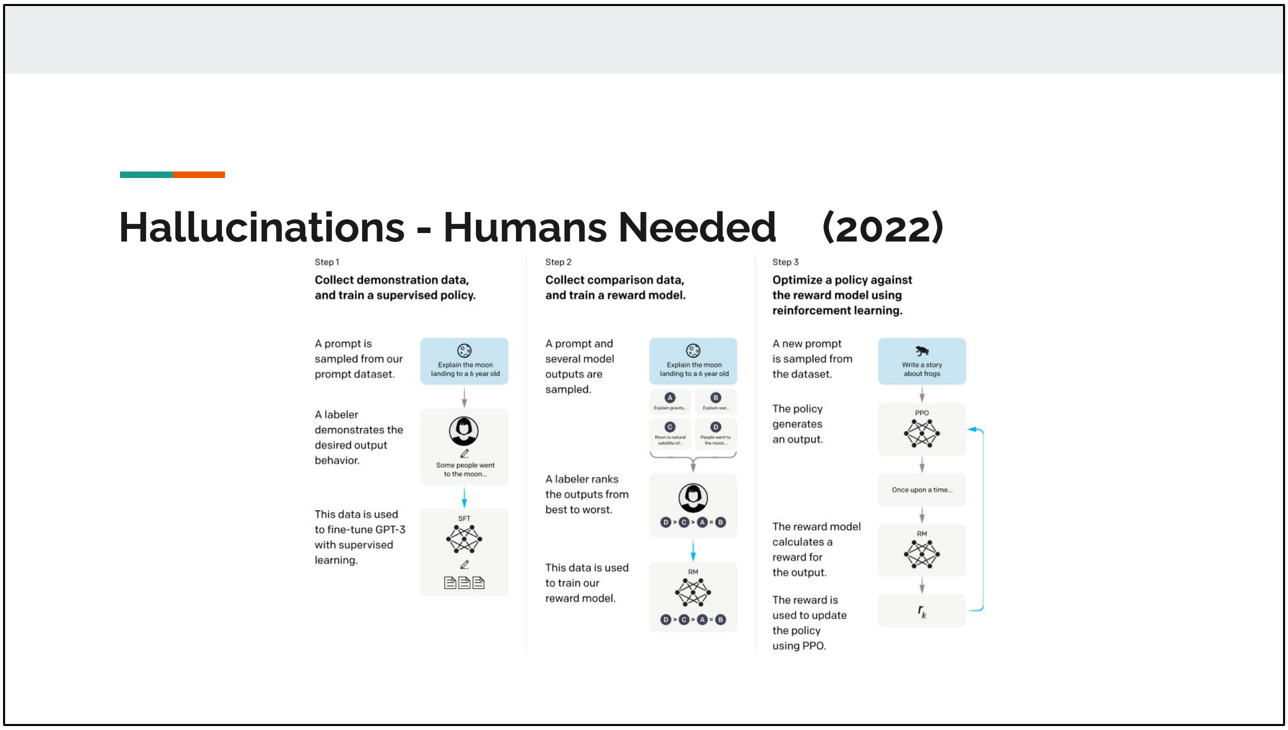
GPT3 175B
This one shown that One Shot or Few Shot learning

This gave Birth to ChatGPT which was the piece of software that got 100 million users the fastest.
And ChatGPT4 which is wild to work with.

Zero Shot And Few Shot

[Anthony Doran] Pretty wild. So as these models have got so big and can do this text to text transform this introduced what we call this phrase is zero shots and few shot.
[Anthony Doran] So you can use the way of interacting with the input to generate your desired outcome. And I think I'll just show you.
[Anthony Doran] I think it's easier to show you this, but you probably already know this. I don't know. Have you guys used the platform open AI. You have okay great. You probably know this when the people referring to zero shot learning. They're talking about scenarios where you instruct the prompt to say like a
[Anthony Doran] couple of examples and I say what's one plus seven equals and I ask it to complete. Are you doing anything. Yeah, so give us an eight great. So I didn't have to give it any examples in you exactly what I was trying to do.
[Anthony Doran] And it knew that in the situation of few shot you're giving it example so that knows how to complete itself. So for example, you might have to help me out here. So the whole thing is the goal is the geese right.
[Anthony Doran] Bride. Lions.
[Anthony Doran] Anyone know of another one. No.
[Anthony Doran] Oh, yeah, school. Okay.
[Christopher Seaman] So I've given it some few shot learning tasks and I'm asking it.
[Anthony Doran] Yeah, well, let's put it backwards. You know, we can just ask it.
[Anthony Doran] What is the plural. What do you call this thing. Word thing. Or collective.
[Anthony Doran] It's a scary. Nice. Okay. Anyway, this thing is ridiculously knowledgeable. And the playground here you get to choose.
[Anthony Doran] Text eventually three is based off GPT three. In GPT four, they

recommend you use the.
[Anthony Doran] But yeah, so any questions about what the difference between zero shot and two shot is. Okay, so that's cool.



Hallucinations - Humans Needed (2022)

Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

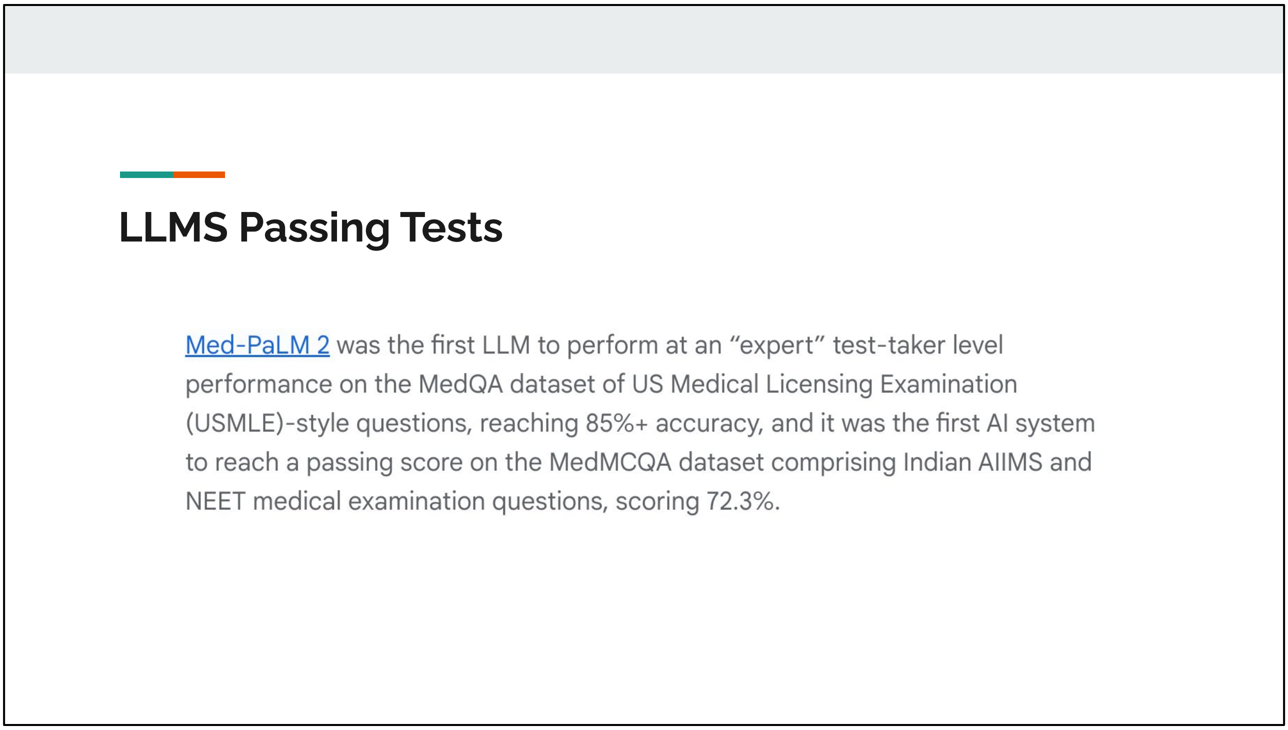
Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

[Anthony Doran] So in 2022. 2022.
[Anthony Doran] Open AI released their paper about how to deal with hallucinations. You've probably heard about hallucinations. It makes the news a lot.
[Anthony Doran] It's like, Hey, this told me, you know. They're paper refers to something called.
[Anthony Doran] Training language models to follow instructions with human feedback. It's referred to as our LHF. If you've heard people say that, that's what they're referring to.
[Anthony Doran] It's basically it's reinforcement learning with the human feedback. And so open AI has a whole bunch of humans that are labeling and annotating data sets. Getting scoring stuff from actual questions to give it weights to then train back into the model to help minimize the limitation minimize the hallucinations chat GPT was released after this with this in mind.
[Anthony Doran] And they're just going to keep on building and removing hallucinations in this way. So that's cool.
[Anthony Doran] What's next. I don't think so. Not that I know.

—

The paper “Training language models to follow instructions with human feedback” happened in 2022

This is what people are referring on RLHF (reinforcement learning with human feedback)

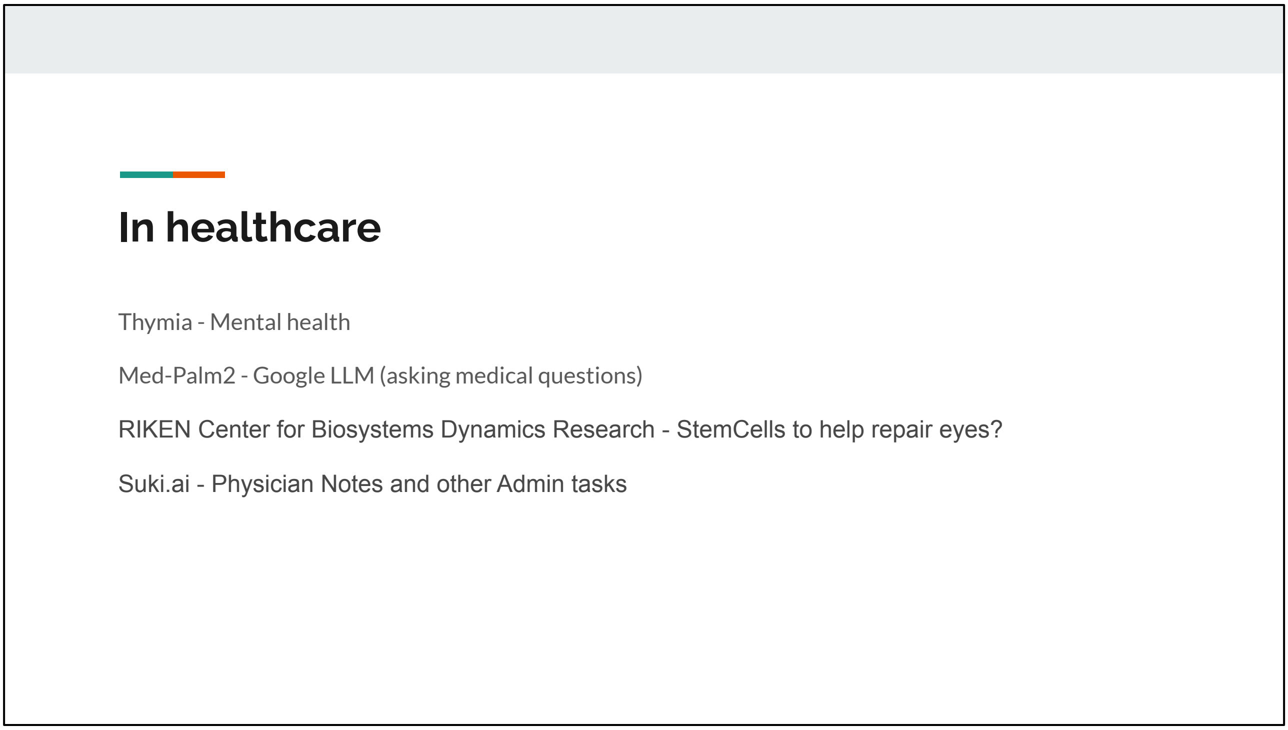


LLMS Passing Tests

Med-PaLM 2 was the first LLM to perform at an “expert” test-taker level performance on the MedQA dataset of US Medical Licensing Examination (USMLE)-style questions, reaching 85%+ accuracy, and it was the first AI system to reach a passing score on the MedMCQA dataset comprising Indian AIIMS and NEET medical examination questions, scoring 72.3%.

[Anthony Doran] All of this is super new or I haven't come across it. Maybe there is. So, LMS are passing tests, which is scary.
[Anthony Doran] There's a medical exam. Have you seen this? Okay. Yeah. So this is palms. It's palms is Google's large language model. It's, it's using a transformer architecture as well, but it's slightly different. They're all slightly different, you know.
—

I saw a blog post from google saying that Palm is scaling to 540 Billion Parameters



In healthcare

Thymia - Mental health

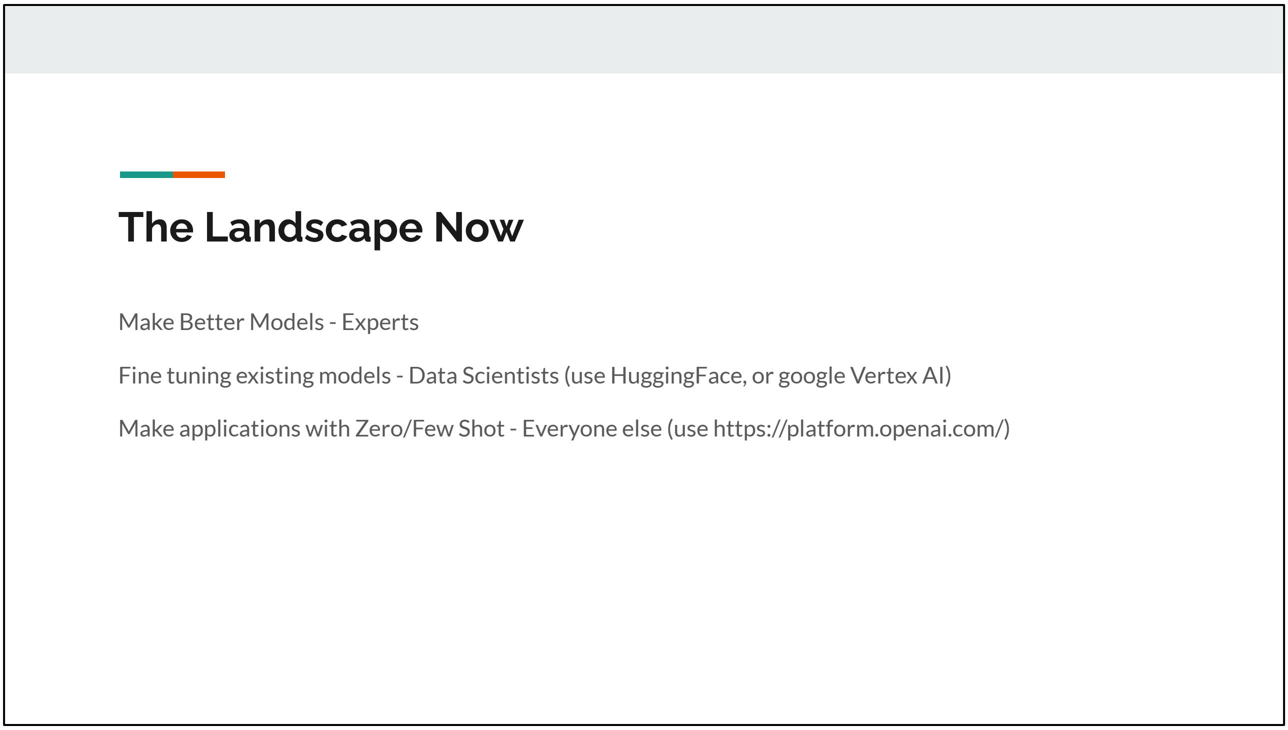
Med-Palm2 - Google LLM (asking medical questions)

RIKEN Center for Biosystems Dynamics Research - StemCells to help repair eyes?

Suki.ai - Physician Notes and other Admin tasks

[Anthony Doran] And so that's a thing. You guys probably know much more than I do, but I just did a quick search for some people using this technology.
[Anthony Doran] It's a crazy one about Fymia that does like based on the sound of your voice. It will try to predict mental health problems.
[Anthony Doran] So you just talk to it. Wild.
[Anthony Doran] So, right in center for this is a Japanese research company doing some sort of weird detection from stem cells and I, you guys would probably know much more from that but that was, that was one that came up. And then just like.
[Anthony Doran] Suki. It's just making a chat interface for our physicians just so that they can talk and keep in their notes in a HIPAA compliant way, which is another one I found.
[Anthony Doran] And I know of any others they're doing like kind of crazy stuff like this. If not, it's cool.

Holy Crap!
Thymia
With just 20 seconds of speech, we can calculate a user's level of mental strain including specific scores for exhaustion, sleep propensity, stress, distress, and self-esteem.



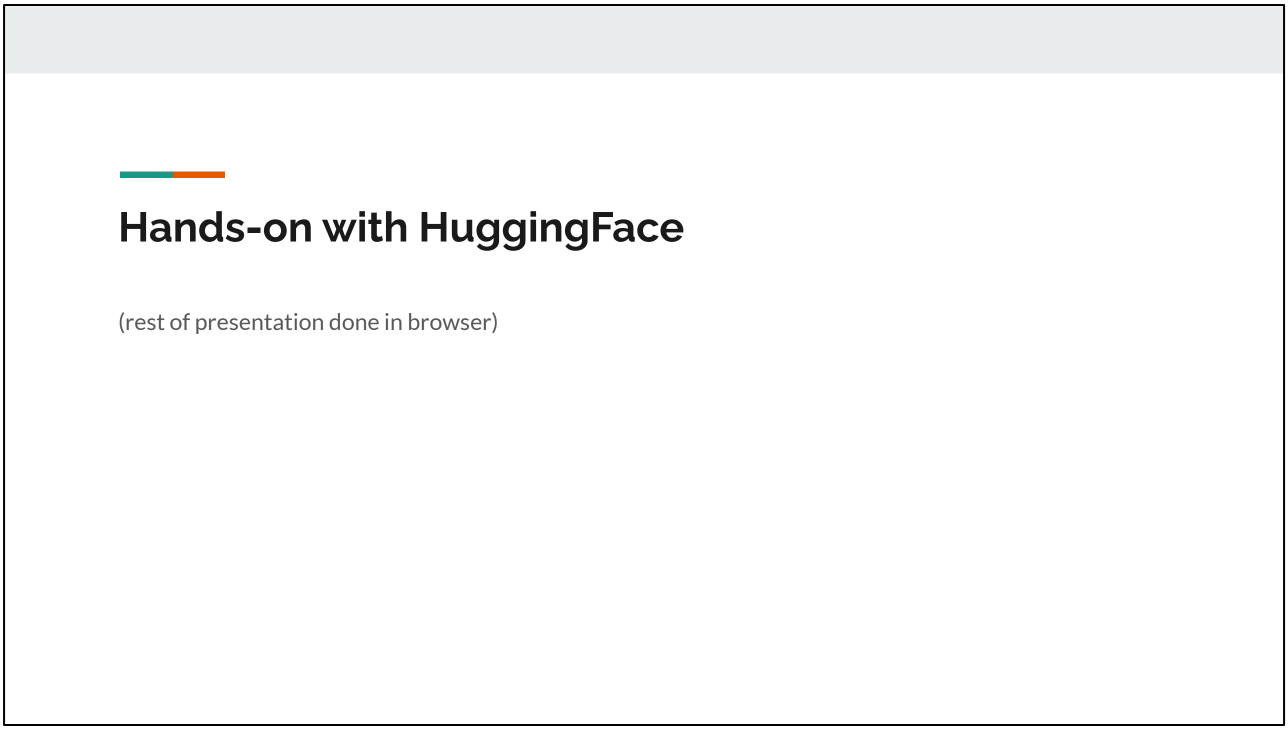
The Landscape Now

Make Better Models - Experts

Fine tuning existing models - Data Scientists (use HuggingFace, or google Vertex AI)

Make applications with Zero/Few Shot - Everyone else (use https://platform.openai.com/)

[Anthony Doran] Keep them to yourself. I recommend it. So this is how the landscape is as of now, if I was thinking about the world of AI.
[Anthony Doran] And then there's these core fundamental language models that the experts and the PhDs are gobbling up all the experts to work on different ways of optimizing them to make them have more information and minimize their hallucinations. Then there's people like us who can find have access to fine tune these models for our specific purposes.
[Anthony Doran] And I'm going to show you an example of that in a second of where to go to get the resources so that you could do this for yourself. Super cool and it's me. And then there's people more like me, where we can make few shot and zero shot learning capabilities to get functionality up there. I'll show you examples with my app.



Hands-on with HuggingFace

(rest of presentation done in browser)

[Anthony Doran] But one of the things I want to show you, which is cool. So I don't know if you guys have heard of hugging face.
[Anthony Doran] Okay, great. Yeah.
[Anthony Doran] Hugging face has all the open source models out there. I think I put a link in there for all the open source models that are there for you to play with. One of the cool things in here is that they have their own course about how to set up your Google Google, Google, Colab notebooks, and interact with this transformer architecture.
[Anthony Doran] Open up the link right here and start interacting and fine tuning whatever you want, whichever model you want on here, which is wild that it's that easy. I mean, you think of all the craziness I just mentioned about like, attentions and networks and stuff like this. And it's a matter of a couple Python code to pull it down from the Internet and then make your own model.
[Anthony Doran] So we, we live in a wild, wild world. Anyway, it was probably what all the old engineers always say for every generation. Anyway. So, so that, that I wanted you to be aware of fun stuff, you know, get a drink and like play with play with models.
[Anthony Doran] And. And then, I could show you.
[Anthony Doran] Oh, yeah, one more thing I wanted to show you. Don't have it up here. So, it's interesting. Right when open AI was making it available for
[Anthony Doran] people to access GPT for the developers, what they did was they said like, Hey, here's our repo of how we evaluate our models. If you want.
[Anthony Doran] The mid PR to help train our model, and we'll give you early access. And then here, gives you these docs about how it evaluates its model and giving you prompt techniques of how to actually make really effective prompts.

[Anthony Doran] So, I'll post this in there later, but yeah, this is a really great resource. Like, so some really funny ones about, given this, this is the state of the chessboard what's the best chess move next to do.
[Anthony Doran] And then, you know, where it's like, given this logical conundrum how do you solve it. You know, and it goes through all these different.
[Anthony Doran] All the different PR so if you go, I don't know if you guys familiar with using GitHub and source control a little bit. Okay. Okay. So yeah, the, the PR is in here will show you examples of what people have done in here.
[Anthony Doran] And type those which ones was seating arrangement. Let's see. This one is like an array of seeing arrangements constraints, each with two solutions examine the models for spatial reasoning.
[Anthony Doran] So it's like giving that that model spatial reasoning capabilities. Let's see another one.
[Anthony Doran] Simple maze puzzles, different geometry puzzles. I think I saw one about.
[Anthony Doran] I have stuff about Japanese. Anyway, neat stuff. Okay.
[Anthony Doran] So, any questions at this point, do you guys have or anything. You need me to shut up for a little bit, because I've been talking a lot.
[Christopher Seaman] No, okay.
[Anthony Doran] Oh, yeah. The power of the uncomfortable silence will it come through.
[Christopher Seaman] No. It's cool.
[Anthony Doran] You want to see how I'm using the prompts. Okay.
[Anthony Doran] So, I'm not an expert in having face, to be honest, I wish I was. I think one, one, one thing.
[Anthony Doran] Is interesting is that this is, this is such a nerdy topic. That the open source community is actually going to become really powerful because of that little, you know, and so we're going to see really powerful large language models generated out of the open source community here.
[Anthony Doran] I mean, faces kind of like the front end for the hard for that open source communities access to all those models.
[Christopher Seaman] In place, some areas kind of like, it for models with.
[Christopher Seaman] Reference models to from the boundaries and then use them, extend them and refine them with additional data. Yep.
[Christopher Seaman] And also offers some including some free ones to run your models on their servers.
[Anthony Doran] Yeah, it's cool. So you don't, it's just, it's just a. Yeah.
[Anthony Doran] For some of the models to use them, you'll need to sign some disclaimers saying like, yeah, I'm not an evil person. I'm not going to, I'm not going to generate a horrible virus. I'm not going to generate, you know, do horrible things. To use them.
[Anthony Doran] We can try. Want to run us some collab books and see what problems are going to. Yeah, let's do that.
[Anthony Doran] Let's learn to get. I have not done this. So we'll see.
[Christopher Seaman] We'll see.

[Anthony Doran] We'll see. Let's go to.
[Anthony Doran] Let's find tuning. Start processing there.
[Anthony Doran] So, pie torches, pythons. I don't know if you guys use pie torch.
[Anthony Doran] Okay, great. Yeah, I give Google, Google collab money.
[Anthony Doran] Yeah, what notebooks do you guys use or notebook? You're using Google collab or you using something else. Are you running the Jupiter locally or do you have any service? Yeah. Okay.
[Anthony Doran] Well, this is cool because in Google collab, we can just attach some GPUs to this and give them, you know, and say like, Hey, be faster. It's lovely.
[Anthony Doran] Sweet.
[Christopher Seaman] Oh, you just put a buff. Oh, you just put a buff.
[Anthony Doran] Let's restart the runtime.
[Christopher Seaman] Let's give it a GP. Why not?
[Anthony Doran] Let's give it a runtime. Yes.
[Anthony Doran] Let's be starting right now. We'll see how good. How well it does. He's boiled.
[Christopher Seaman] Bold.
[Anthony Doran] I can catch some hardware to it. Yeah.
[Anthony Doran] Oh, yeah, the other thing about in hugging faces that you have access to all a bunch of data sets that actually becomes really useful as well. Like they have examples of chat interfaces about asking questions and getting answers so that you can make sure that like you have your model is scoring above a certain number for that.
[Anthony Doran] You know, there's the classic example of all the pictures of all the handwritten notes. You know, and then their numbers or handwritten numbers and then the.
[Anthony Doran] That kind of stuff is on there. Cool. Right. There's this thing about checkpoints.
[Anthony Doran] I can't remember what they are. Okay.
[Anthony Doran] Well, the notebook ran. That's good.
[Anthony Doran] Before, whatever. I don't know what kind of use cases you guys are imagining of what you would do to interact. But before you got into the land of training the model for a specific sub case.
[Anthony Doran] I would double check first. If a few shot.
[Anthony Doran] If using a few shot on the large language model will actually satisfy whatever your need is first. You know, because it'll save you a bunch of time.
[Anthony Doran] Okay, so let's let's think of example what I mean, I don't know. Do you guys have examples of something you think a language model would actually be useful for. You can think of it. It's hard to kind of hard to think, you know,
[Christopher Seaman] a few shot on the large language model on.
[Anthony Doran] That was straight on one thing that I want to apply some day more specific. Yeah, exactly. So I don't know. Let's say.
[Anthony Doran] Let's say you had examples of like. You have a series of skin problems that people put into language and you want to get a potential like the top reason that it might be what what it might be is the problem.

[Anthony Doran] What I would first do is go into the few shot examples and say like patient says this. And if they say this, this is the problem that they have.
[Anthony Doran] And then it says this, I'm just generating a very small.
[Christopher Seaman] Yeah, tell it. Give it a prompt. Yeah. Yeah. Like this is the disease and the other.
[Anthony Doran] Yep. And then, and then, and then we can try to just do five of them to see how well it does first. And if it's doing pretty well, you know, push it a little further and see where it falls apart.
[Anthony Doran] And then you can see down. The training, you know, just to see how well it goes. And there's a bunch of circumstances like this, it's, because it's.
[Anthony Doran] It's amazing what's already baked into the chat GPT for model. Our tendency is to go for the nerdiest thing possible as quickly as possible, but like it, it handles it.
[Christopher Seaman] And then you can have other examples of things that you think will be. Even in general.
[Christopher Seaman] Train on general text and apply it to. Specific, I can buy it to something medical.
[Anthony Doran] Start off. Or even more specific.
[Anthony Doran] Yeah. Yeah.
[Anthony Doran] It's wild to see it. So I'd start with the few shot first.
[Anthony Doran] Or even the zero shots like hey well problems this, you know, and then work your way back up. There might be reasons you have to go to fine tuned first because of some sort of like.
[Anthony Doran] You need a certain assurance that these are all good. There's this really cool technique in this.
[Anthony Doran] In this. In open AIZ valves.
[Anthony Doran] And the way to make sure to let the to ask the model. To explain why it answered things a certain way. And also to give it a sense of how good it thinks its answers are.
[Anthony Doran] And it looks like this. So instructions.
[Anthony Doran] They have this model graded eval template. So if you have.
[Anthony Doran] Let's say you have a phrase like hey my skin has been itchy for the last 10 days. And it's supposed to be.
[Anthony Doran] Hey, you just got acne or something. I don't know, you know, that's what it's trained to be the answer for. And, and then you ask open A.I.
[Anthony Doran] To prompt it. And what you.
[Anthony Doran] Can do is then after the output to see the answer and see what it got right or wrong. You can ask it to say like, how did you get to that conclusion?
[Anthony Doran] And do you think your answer is right or wrong. So you can feed it back in.
[Anthony Doran] And it'll actually do a really good job of explaining its reasoning for why it answered things a certain way. You know, and there's a bunch of ways to automate that process to actually make sure that it's actually your your model is actually performing well.
[Anthony Doran] Just neat.

[Christopher Seaman] So, we like where do we. We have like a disease.
[Christopher Seaman] And it has like four based on how severe the business is. And so it has like four classifications.
[Christopher Seaman] We paid that until like we. We gave it some information like these are the symptoms or like signs.
[Christopher Seaman] We know after we get to know where it is lacking.
[Anthony Doran] Yeah, great. Great. Yeah, it's a great question. I am. This is where the fine tuning comes in.
[Anthony Doran] I'm not an expert in it. We will follow the docs that that exists on hugging face for that, to be honest, you know.
[Anthony Doran] But you got to get to that point first, you know, I mean, you know, once you get to that point, then you can start doing. Yeah.
[Anthony Doran] One of the. Let me show you something that's actually quite interesting.
[Anthony Doran] Of how I interact with it with in my product into in that example. It's a good example. Let's see.
[Anthony Doran] So.
[Christopher Seaman] We put on so like. Yeah.
[Anthony Doran] And like. Oh, man, do you have patient notes? We can just try that out right now.
[Anthony Doran] Oh, that'd be. That'd be awesome. That'd be great. I think it would do really well with something like that. Yeah.
[Anthony Doran] There must must be some sort of legal loopholes to get through to make sure you that you can do that. I know.
[Anthony Doran] But it would be fascinating to see if you can anonymize it well and then put it through.
[Christopher Seaman] It'd be really great to try that out. Download models from.
[Christopher Seaman] Oh, yeah, I bet they do have a model that you wanted to use on hugging faces of face. So, you know,
[Christopher Seaman] or by. And you shop learning.
[Christopher Seaman] So give it some prompts and then prompts in context, which is a kind of. So use that in an offline model that you train.
[Christopher Seaman] Based on something open source in how you face. And not worry about sending.
[Christopher Seaman] Help information to open AI.
[Anthony Doran] Yeah. Probably be at least in past non-brother running. Yeah.
[Anthony Doran] Right. So, just, I'm not surprised to find that.
[Christopher Seaman] How would you choose from all of these? If you just saw this, how would you start to know it and you didn't get it.
[Anthony Doran] So, let's go in and take a look. So here we have. They, they usually pretty well documented about what each data set has.
[Christopher Seaman] Well, what about the basics, like how many people are using this or like it. Like if I want to just get a blank.
[Christopher Seaman] I don't know. Yeah.
[Christopher Seaman] There's a sense of community around it. Yeah.

[Anthony Doran] Probably do. Here's a list of models trained on those questions.
[Christopher Seaman] So that's what I'm going to get for.
[Anthony Doran] Yeah.
[Christopher Seaman] Yeah. True.
[Anthony Doran] Let's see.
[Christopher Seaman] Let's see. Let's see if we can get like.
[Anthony Doran] Here we have downloads. Yeah.
[Anthony Doran] Looks like it's, this is the top one up there. Oh, no question pairs.
[Anthony Doran] I want to show you how crazy you can actually put stuff into the prompt. So we build an internal tool simply put where we allow people to connect to their database.
[Anthony Doran] And then we make a chat bot that allows them to ask natural language questions over their data. And I can just talk to it.
[Anthony Doran] And it's basically generates text is equal, but it does a lot more than that. And this is the prompt right here that we give open AI to do the translation.
[Anthony Doran] And what we, when we were first developing this, we say like, here's the prompt. I'm going to do you to, to open AI.
[Anthony Doran] We told open the prompt and say, Hey, can you make this efficient as possible open AI so that you understand it better. And we had it correct itself, a bunch of correct or prompt for it.
[Anthony Doran] A bunch of times. And here in this circumstance, we give it chapters of which it likes chapters and chapters of rules.
[Anthony Doran] And in situation, and we're going to generate some JSON at the end of this so that we can parse it. And we tell it what should be in each of the JSON fields.
[Anthony Doran] And so for example, provide a summary field with a single sentence summary, if the SQL query results to see a single row. So that we don't have to do that. Right. And you can give it a whole series of rules. We give it a couple of examples of what the output is.
[Anthony Doran] And, yeah, and there's a bunch of other subtleties in here that's actually interesting that you can then really flush out your prompt engineering programmatically to then interact with it, which is. So it's the rules you can get in here. We also allow our data people are data admins who are monitoring this data to put in their own rules into the prompt, which allows for.
[Anthony Doran] Some neat experiences where they have some hidden knowledge over like, Oh, this table needs to be joined this way. You know, or, Oh, this table actually represents this, even though it's named this way.
[Anthony Doran] You know, and give it hints and stuff. But yeah, you can actually put in a bunch of different logic in here and it'll understand what to do.
[Anthony Doran] Just pretty well, and this seems like a really big prompt. But yeah, so, yeah, I just wanted to show you that.
[Anthony Doran] Like I wouldn't. You know, it's not as cool Python, you know, by going really into the model, but you can do a lot with prompt engineering is my experience.
[Anthony Doran] It's actually a very, very powerful tool in your tool in there.
[Christopher Seaman] Cool.

[Anthony Doran] Not that I know, because it's so new. You know, this is like, and there's guys are going to have video YouTube videos claiming bunch of stuff you have to be kind of careful, right.
[Anthony Doran] I don't know. Do you guys learn a lot from YouTube?
[Christopher Seaman] Yeah. Yeah.
[Christopher Seaman] It's better to leave. Oh, man, I mean, the amount of music knowledge I've learned from YouTube has been amazing.
[Anthony Doran] It's been amazing for my piano. It'll actually be really good for this nerdy stuff too, about getting into prompting you.
[Anthony Doran] Yeah. Oh yeah.
[Anthony Doran] Yeah. Raren.
[Christopher Seaman] I do.
[Anthony Doran] Are they good that you got good content?
[Christopher Seaman] Yeah. Right.
[Anthony Doran] Well, this topic in particular is so new. I mean, it was really the beginning of this year. So it's faster, almost to see people just like, Oh, I'm so excited about this. Let me post a video.
[Anthony Doran] Then we have like the well written document around stuff.
[Christopher Seaman] Mostly, but like anything you've seen it applied on the video. So a lot of this is getting applied in places that are heavy already developed in their own.
[Christopher Seaman] So Microsoft Google meta will have their own prompt engineering understanding, but that will necessarily be applied to medical data. Whatever is not directly relevant to them.
[Anthony Doran] Yeah, it's funny how the culture of the industry will really affect the adoption here.
[Christopher Seaman] Yeah. Which makes for an opportunity for a very basic, what they might do want to have to do.
[Christopher Seaman] And then they'll be able to do it.
[Anthony Doran] Yeah, it's gonna be wild. Cool. That's all I got. I could show you cool stuff I can do with my application, but I don't know if that's actually useful to you guys.
[Anthony Doran] I am. We, the current problems our company faces right now is staying up with the innovation in the world. And being able to make sure we're using the best model that's out there.
[Anthony Doran] Because that changes quickly. Like it's turning out that bard is turning out to be really good. As an L alone.
[Anthony Doran] It's just not gotten the publicity of the GPT for, but this is going to change. And one of the, one of the things our, our apps is.
[Anthony Doran] Struggles with is the response time from asking a question and getting that. So we're looking to get help.
[Anthony Doran] Either doing stuff internally. And then we're doing it ourselves versus being so dependent on an external source.
[Anthony Doran] Anyway, these are our current challenges. But it's cool.
[Anthony Doran] Yeah, I don't know. You guys want to talk about anything else or other than that. That's what this whole I got.
[Anthony Doran] You know, or I should probably stop the video at this point.

[Christopher Seaman] Yeah.