GPT generated essay

Large language models have become a hot topic in machine learning research in recent years. These models are trained through self-supervised training, which involves autoregressive training and next token prediction. The pre-training is done through either masked language modeling (MLM) or causal language modeling (CLM). The models are trained on any kind of text, including code completion and tabular data completion. The pre-training is done on a large scale, and the models are versatile and can be adapted to new tasks quickly.

However, the foundational models are not usable in their raw form. They need to be fine-tuned for specific tasks. There are two main methods of fine-tuning: supervised fine-tuning and reinforcement learning with human feedback. Supervised fine-tuning involves creating a diverse set of prompt data sets compatible with the task, showing them to labelers, and then feeding the query responses into the model for fine-tuning. Reinforcement learning with human feedback involves injecting a good kind of bias to the models.

Large language models have been developed in deep learning, and the majority of the development has been on transformer-based models. These models are mostly trained through self-supervised training, and they are versatile and can be adapted to new tasks quickly. The pre-training is done on a large scale, and the models are not usable in their raw form. They need to be fine-tuned for specific tasks, and there are two main methods of fine-tuning: supervised fine-tuning and reinforcement learning with human feedback.

In a presentation given by Ehsan Kamalinejad, large-language models and machine learning research were the main topics. He discussed the process of creating responses for specific domains and how to create a diverse setup for open AI. The responses are then exposed to the model, and the labelers rank them based on their quality. The ranking mechanism is used to create a rewarding rule, which identifies better and worse responses based on previous examples. The reward model is then applied to every enforcement learning to the base model, which creates a text, and the reward model determines whether the answer is good or bad.

Kamalinejad also discussed the difference between supervised fine-tuning and RL HS. Supervised fine-tuning involves crawling data from the internet, while RL HS requires online data collection. RL HS is preferred because it requires less effort from labelers, but supervised fine-tuning is easier to collect data for. However, supervised fine-tuning suffers from catastrophic forgetting, where the model forgets some of the knowledge from the past. RL HS suffers less from this phenomenon, making it a better option.

Kamalinejad also demonstrated how supervised fine-tuning changes the behavior of the model. He fine-tuned a llama model on a modest-sized dataset and showed how it behaved differently after fine-tuning. He also discussed his free course on RL HS and how it explains all the pieces of the model.

Large-language models have become increasingly popular in machine learning research, but there are still many questions that need to be answered. One of the main questions is how much data is needed to fine-tune a language model to produce good results in a specific domain. While there is no agreed-upon answer, empirical results suggest that tens of thousands of samples are needed for good fine-tuning. This is a relatively small amount compared to the general data sets required to train deep neural network models.

Another important question is how much improvement can be achieved through fine-tuning, and what metrics can be used to measure the quality of these models. Traditional benchmarks in language are no longer applicable, and new metrics are needed to measure the effectiveness of these models.

Scaling laws are also important to consider when applying limited compute to training. The Chinchilla paper from Deepmind provides some insight into scaling laws for pre-training, but the question of fine-tuning is still vital.

One interesting use of large language models is synthetic data generation. Kamalinejad suggests that companies can collect enough data, throw it at OpenAI API, and get a bunch of training sets that can be used on small models. However, there are licensing and legal issues with using OpenAI API for training your own models. Kamalinejad recommends sticking to the principles of fine-tuning and not marrying to any specific method. He suggests that reinforcement learning (RL) is a very generic method that can be applied to any model.

Kamalinejad also emphasizes the importance of data in machine learning models. He notes that large language models are powerful because they were trained on data that most other people don't have. AWS serves cohere models and has a new initiative to create foundational models and collaborate with other labs to make open source models more widely available.

In conclusion, large language models have revolutionized machine learning research, and companies can benefit from fine-tuning them with open source data and using labeling tools for data collection. Synthetic data generation is also a useful technique, but there are licensing and legal issues to consider. Sticking to the principles of fine-tuning and emphasizing the importance of data are crucial for success in machine learning.

Q and A format

What are large language models and how are they trained?

Large language models are a type of machine learning model that are trained through self-supervised training, which involves autoregressive training and next token prediction. The pre-training is done through either masked language modeling (MLM) or causal language modeling (CLM). The models are trained on any kind of text, including code completion and tabular data completion. The pre-training is done on a large scale, and the models are versatile and can be adapted to new tasks quickly.

Why do large language models need to be fine-tuned and what are the methods of fine-tuning?

The foundational models of large language models are not usable in their raw form. They need to be fine-tuned for specific tasks. There are two main methods of fine-tuning: supervised fine-tuning and reinforcement learning with human feedback. Supervised fine-tuning involves creating a diverse set of prompt data sets compatible with the task, showing them to labelers, and then feeding the query responses into the model for fine-tuning. Reinforcement learning with human feedback involves injecting a good kind of bias to the models.

What is the difference between supervised fine-tuning and reinforcement learning with human feedback?

Supervised fine-tuning involves crawling data from the internet, while reinforcement learning with human feedback requires online data collection. Reinforcement learning with human feedback is preferred because it requires less effort from labelers, but supervised fine-tuning is easier to collect data for. However, supervised fine-tuning suffers from catastrophic forgetting, where the model forgets some of the knowledge from the past. Reinforcement learning with human feedback suffers less from this phenomenon, making it a better option.

What are some important questions that need to be answered about large language models?

One of the main questions is how much data is needed to fine-tune a language model to produce good results in a specific domain. While there is no agreed-upon answer, empirical results suggest that tens of thousands of samples are needed for good fine-tuning. Another important question is how much improvement can be achieved through fine-tuning, and what metrics can be used to measure the quality of these models. Traditional benchmarks in language are no longer applicable, and new metrics are needed to measure the effectiveness of these models. Scaling laws are also important to consider when applying limited compute to training.

What is an interesting use of large language models?

One interesting use of large language models is synthetic data generation. Companies can collect enough data, throw it at OpenAI API, and get a bunch of training sets that can be used on small models. However, there are licensing and legal issues with using OpenAI API for training your own models. Kamalinejad recommends sticking to the principles of fine-tuning and not marrying to any specific method. He suggests that reinforcement learning (RL) is a very generic method that can be applied to any model.

What is the importance of data in machine learning models?

Kamalinejad emphasizes the importance of data in machine learning models. He notes that large language models are powerful because they were trained on data that most other people don't have. AWS serves cohere models and has a new initiative to create foundational models and collaborate with other labs to make open source models more widely available.

Interview Format

Interviewer: Welcome, everyone. Today we have Ehsan Kamalinejad with us to discuss large language models and machine learning research. Ehsan, thank you for joining us. To start off, could you explain the process of fine-tuning large language models for specific tasks?

Ehsan Kamalinejad: Thank you for having me. Fine-tuning large language models involves taking the foundational models, which are not usable in their raw form, and adapting them to perform specific tasks. There are two main methods of fine-tuning. The first one is supervised fine-tuning, where we create a diverse set of prompt data sets that are compatible with the task. These prompts are then shown to labelers, and their responses are fed into the model for fine-tuning. The second method is reinforcement learning with human feedback, which involves injecting a good kind of bias into the models. This process requires creating a rewarding rule based on the ranking of responses provided by labelers. The reward model is then applied to every enforcement learning to the base model, which generates text, and the reward model determines the quality of the answers.

Interviewer: Could you elaborate on the difference between supervised fine-tuning and reinforcement learning with human feedback?

Ehsan Kamalinejad: Certainly. Supervised fine-tuning involves crawling data from the internet, which requires collecting a significant amount of labeled data. On the other hand, reinforcement learning with human feedback requires online data collection. While supervised fine-tuning is easier to collect data for, it suffers from a phenomenon called catastrophic forgetting, where the model forgets some of the knowledge it learned from the past. Reinforcement learning with human feedback, or RL HS, suffers less from this issue, making it a better option. RL HS also requires less effort from labelers, which is another advantage.

Interviewer: During your presentation, you mentioned the behavior changes in a fine-tuned model. Could you explain this further and provide an example?

Ehsan Kamalinejad: Certainly. I demonstrated how fine-tuning a llama model on a modest-sized dataset can change its behavior. The model behaves differently after the fine-tuning process, adapting to the specific task it was fine-tuned for. For example, if we fine-tune a language model for customer support, it will learn to generate responses that are more suitable for that domain.

The fine-tuning process allows us to customize the behavior of the model to better fit our desired application.

Interviewer: One of the questions raised in your presentation was about the amount of data needed for fine-tuning a language model. Could you provide some insights into this?

Ehsan Kamalinejad: Certainly. While there is no universally agreed-upon answer, empirical results suggest that tens of thousands of samples are needed for good fine-tuning. This amount of data is relatively small compared to the general datasets required to train deep neural network models. However, it's important to note that the specific requirements may vary depending on the complexity of the task and the desired performance of the model.

Interviewer: Another important question you mentioned is how much improvement can be achieved through fine-tuning and what metrics can be used to measure the quality of these models. Could you elaborate on this?

Ehsan Kamalinejad: Absolutely. Traditional benchmarks in language are no longer applicable to measure the effectiveness of large language models. Therefore, new metrics need to be developed to assess the quality of fine-tuned models. Determining the improvement achieved through fine-tuning depends on the specific task and the desired performance metrics. It's an area of ongoing research to establish appropriate evaluation methods that capture the capabilities and limitations of these models effectively.

Interviewer: Scaling laws were mentioned as an important aspect when applying limited compute to training. Could you explain what scaling laws are and their relevance to fine-tuning?

Ehsan Kamalinejad: Scaling laws refer to the relationship between computational resources and the performance of machine learning models. In the context of large language models, scaling laws provide insights into how increasing computational resources affect the pre-training process. While the Chinchilla paper from DeepMind provides some insights into scaling laws for pre-training, the question of fine-tuning is still vital. Understanding the scaling laws for fine-tuning can help optimize the allocation of limited compute resources for training large language models.

Interviewer: Synthetic data generation was mentioned as an interesting use of large language models. Can you explain this concept and its potential benefits?

Ehsan Kamalinejad: Synthetic data generation involves using large language models to generate artificial data that can be used for training smaller models. The idea is that companies can collect enough data, throw it at the OpenAI API, and obtain a bunch of training sets to use with their own models. However, it's important to consider licensing and legal issues associated with using OpenAI API for training proprietary models. Despite these concerns, synthetic data generation can be a valuable technique for augmenting datasets and improving the performance of smaller models.

Interviewer: You emphasized the importance of data in machine learning models. Could you elaborate on this and how it relates to large language models?

Ehsan Kamalinejad: Data is crucial in machine learning models, including large language models. Large language models are powerful because they have been trained on vast amounts of data that most other people don't have access to. The availability of comprehensive and diverse datasets enables these models to learn patterns and generalize well. It's important to consider the quality, diversity, and relevance of the data used for training and fine-tuning large language models to ensure their effectiveness.

Interviewer: In conclusion, what are the key takeaways regarding large language models and their application in machine learning research?

Ehsan Kamalinejad: The key takeaways are that large language models have revolutionized machine learning research, providing versatile tools that can be quickly adapted to new tasks. Fine-tuning these models with specific data and utilizing labeling tools for data collection can significantly enhance their performance. Synthetic data generation is an interesting technique to explore, although it's essential to address licensing and legal concerns. Sticking to the principles of fine-tuning, such as reinforcement learning, can yield effective results across various models. Finally, emphasizing the importance of data and collaboration in making foundational models more widely available will contribute to the advancement of machine learning as a whole.

Interviewer: Thank you, Ehsan, for sharing your insights on large language models and machine learning research. It has been a pleasure to have you here.

Ehsan Kamalinejad: Thank you for having me. It was my pleasure to discuss these topics with you.

Multiple choice quiz

Q1. What is the process of fine-tuning in large language models?
a. The process of extracting knowledge from the models
b. The process of pre-training the models
c. The process of training the models on any kind of text
d. The process of creating responses for specific domains

Answer: a. The process of extracting knowledge from the models

Explanation: Fine-tuning is the process of extracting knowledge from the models. It involves creating a diverse set of prompt data sets compatible with the task, showing them to labelers, and then feeding the query responses into the model for fine-tuning.

Q2. What are the two main methods of fine-tuning large language models?
a. Supervised fine-tuning and unsupervised fine-tuning
b. Reinforcement learning with human feedback and unsupervised fine-tuning
c. Supervised fine-tuning and reinforcement learning with human feedback
d. Supervised fine-tuning and semi-supervised fine-tuning

Answer: c. Supervised fine-tuning and reinforcement learning with human feedback

Explanation: There are two main methods of fine-tuning large language models: supervised fine-tuning and reinforcement learning with human feedback.

Q3. What is the preferred method of data collection for reinforcement learning with human feedback?
a. Crawling a lot of data from the internet
b. Online data collection
c. Collecting data from open source models
d. Collecting data from labeled datasets

Answer: b. Online data collection

Explanation: Reinforcement learning with human feedback involves online data collection, which is preferred because it requires less effort from labelers.

Q4. How much data is needed to fine-tune a language model to produce good results in a specific domain?
a. Thousands of samples
b. Millions of samples
c. Billions of samples
d. It depends on the size of the model

Answer: a. Thousands of samples

Explanation: While there is no agreed-upon answer, empirical results suggest that tens of thousands of samples are needed for good fine-tuning.

Q5. What is the main advantage of reinforcement learning with human feedback over supervised fine-tuning?
a. It requires less effort from labelers
b. It is easier to collect data for
c. It suffers less from catastrophic forgetting
d. It produces better results in specific domains

Answer: c. It suffers less from catastrophic forgetting

Explanation: One advantage of reinforcement learning with human feedback over supervised fine-tuning is that it suffers less from catastrophic forgetting, which is a common problem in transfer learning.

Raw Transcript

Ehsan Kamalinejad: Okay, thanks. Amir and thank you everyone for listening. We in fact with Amir, we had a few different topics that we wanted to cover and we didn't know which one to pick for this conversation. I mean, the field is moving so fast and there are so many interesting things but at the end of the day, a few days ago, I decided that Fine-tuning probably is one of the most pressing one. I know that Suhas if I pronounce his name correctly was also talking about this, This is a slightly different angle so there will be a little bit of overlap,… Robert James: Right. Ehsan Kamalinejad: but mostly different type of Material. So with that, let's start about talking about fine-tuning, large language models. And the base idea is that How do you create customized models that are specific for your domain, or your problem? Okay. Ehsan Kamalinejad: Let's just start with a very quick history of the development in large language models. So, this is, in fact, a tree that I borrowed from. I leave all of the links in the slides so you can later take a look at them. So these are different types of language models that have been developed in deep learning so we don't go before deep learning. Of course, there are like other stuff too, but in the modern deep learning era, these are mostly the kind of development that we have. So, in particular, we're gonna take a look at only the transformer based models. And if you, if you narrow our vision only to those models, then we will see these these kind of development in from 2018. So, Ehsan Kamalinejad: And the three, the subtree that you see on the on the most right is the As you can see, it has many leaves. Ah, that's the decoder. Only sub problem. And these problems are problems that these models and tasks are the auto regressive tasks. Basically, the the pre training of these models is through next token, prediction mostly. There are variations of it but it's mostly like that and it's essentially it started with GPT. So open AI. I started Ehsan Kamalinejad: That that approach. There's also the other parallel approach in encoder decoder. Probably, you have seen flan and family of models from Google, those work very well too. For some problems, but I can save it high confidence that the majority of the development at least in the past two years has been on this on this right. Subtree. And you can, you can see all like Bart, Gpt4lama blah, Ehsan Kamalinejad: And so, these models are mostly trained to self-supervised training. So either through maths language modeling MLM, as hugging phase, call it or causal language modeling again, clm as hugging face, Ah, call it. I know that Amir doesn't believe that these are actual causal models. We have a little bit of disagreement over there with Amir, but at least on the terminology we call them puzzle language models. So these are the models that basically get some context and create some responses for that context. In that sense we call them causal. So this is a little bit of history of the development in this domain. And as you can see, it's very hot. A lot of development. so, Ehsan Kamalinejad: how does the parade training is done? It's as I said, it's a pretty simple. Simple pre-training and the idea is pre-training is a autoregressive type of training and essentially you do next. Next token prediction The good thing about this is that this is this is a very simple setup. First of all simplicity is really important. As you can see, like Python language is not necessarily the best programming language. But since it's simple, it has flaw like flying over all of the other languages right now. So it's a simple setup. That's the first thing that is important for it. Ehsan Kamalinejad: More importantly, this allows you to train on any kind of text because if you think about it language, for example, as we have an example here, the chicken walked across, and then you have to fill in the rest of it is, one example from language, but you can do it for other things too. You can do code completion, you can do even tabular data.com and completion, you can do completion for any kind of data that is tokenizable. And we learned in the past few years. For example, most suite recently, by the cosmosphere paper from Microsoft. That's pretty much everything can be tokenized. So, this is a very general purpose, kind of pre training that you can go over it and just train and you can see that in that setting pretty much the whole Internet is your playground and playground that is 00:05:00 Ehsan Kamalinejad: Honestly important because you know these large language models are in fact in general, deep learning models, they get to good results, only at a scale. So this is great. Also the task. It's this something very interesting about the task. All the setup of this task is seen Paul. The task itself is not that simple. It means that it pushes the model to learn very interesting patterns for example, if I give two plus two is equal to and ask for the net and next token as we probably it has been done. Many times in the in the data from the Internet, the model should kind and understand how the mathematics works. and if you say that the president of the US in 1983 was, Ehsan Kamalinejad: What is the next token? The model should know history. So these are just two examples. You you can extrapolate from this. These models have to learn a lot to be able to do this autoregressive task very well. So this is about the pre training. So then you do the pre-training and then you're done with the large language models that you have trained. The next you're not done, in fact the foundational model are trained in large data in, for example, in this case, in the this autoregressive fashion, but in this raw form, Ehsan Kamalinejad: It's they're not that usable they are versatile in a sense that they adopt to a new task. If you basically put a new task on top of these language models, they they can be trained very quickly. You can also do few shot learning on top of them. For example, context, learning in a sense that you have seen before you give some context and you show some examples to the model and then the model follows that And, but in the raw form, they do not follow instructions and they don't answer questions. So the knowledge is stored in them but one needs to extract the knowledge. A very nice diagram of this that I am. Probably some of you have seen is this monster of a language that has like, Ehsan Kamalinejad: All these crazy things going on about it and then you do it and you do certain supervised fine-tuning on top of it and it gets a little bit better. And then you do, finally some reinforcement learning with human feedback. And then this is my league, a cute guy comes out of it. So the way that I like to see, I like to think about it is that you have this rock that in its original form. It's not that shiny. It's not that that beautiful, but you can cut it, you can cut it. And after you do a lot of cutting and you, you make it clean and shiny, then it turns into that diamond that glows and like it. Ehsan Kamalinejad: It's very beautiful, right? So the same idea here. It's not necessarily injection of new knowledge, but it's extraction of the knowledge in these language models, that is done through fine-tuning. And there are two main methods of fine-tuning. There's supervised fine-tuning and there's a Reinforcement, learning with human feedback. Fine-tuning so, RL chef So, yeah, I mean you mentioned, it's essentially you want to inject a good kind of bias to the, to the malls. Um okay so let's quickly think about the supervised fine-tuning. The supervised fine-tuning works like this. You first create a diverse set of prompt data set compatible with your task? Maybe you you care about financial stuff then your prompts probably are focusing in financial material. 00:10:00 Ehsan Kamalinejad: Sample, the prompts data set, and then show these prompts to the labelers. The labor layers, go ahead and complete those prompts with the query appropriate for those tasks, and then this will be the, this will create for you. There's pairs of query responses and then you feed these into the model and you fine-tune the model you still do the fine-tuning in a sense of auto regressive. But now in this specific format that you have query and responses and it's a lot nicer than just crawling the Internet, so it's kind of focused the model into a specific type of interaction. Ehsan Kamalinejad: That's supervised fine-tuning in a nutshell. There are many things about that. I will explain a few of them later. For example, there is the question of efficiency and different types of training. For example, have a parameter efficient fine-tuning, and those kind of things. There's a lot going on here. Ehsan Kamalinejad: At the other type, that is quite different than the previous type is the, then you do reinforcement learning to find tune your language. And there are also several different types of reinforcement fine-tuning. For example, if you're familiar with anthropic constitutional paper, they use other models to do reinforcement learning. So reinforcement, learning with human feedback, human is not necessary in this setup. You can sometimes replace it with another AI. You can also replace it with some specific type of data, but the simplest setup that we have is the RLS chef So, how does the rlhf works? The Arnold shift works like this? Ehsan Kamalinejad: You show some you create the responses again for your specific domain and sometimes if you're open AI, you want to serve everyone. So you don't specify it. You just try to create a very diverse setup. Prompts And then you expose these problems to your model and you get back some responses from the model, then you show these responses together with the question to labelers and the labelers just now rank these responses. So they say that, for example, be the answer like the question can be explained the moon landing to a six years old and then there are four different answers that are created from the model and the labeler comes in and says that answer these the best and then the answer, see, and then a and b are more or less at the same level. Ehsan Kamalinejad: so, the way that the data is labeled is to this ranking, mechan Ism. Then the ranking mechanism is used to create a rewarding rule. The Reward Model In general. What it says, is that it can look at the response and based on the examples that it that it has seen before, from the labelers, it can identify what responses are better and what responses are worse Ehsan Kamalinejad: What you do finally is that now that you have trained and have a reward model, you apply that reward model to every enforcement learning to your basement. Basically the base model creates a text and the reward model says that oh that was a bad answer, Don't do that. This was a good answer. Keep doing that so on and so forth. So, that's in a nutshell, how the, how the training here works so, Ehsan Kamalinejad: There's a lot here though like this one of the reasons that RL Hsf has not lied as much as supervised fine-tuning is that? In fact, this setup is pretty complex. I think this is, this is slides. Probably is the most comprehensive a slides about our electric. It's from colossal AI. These guys are the UC Berkeley Reinforcement learning team. They also work on other domains of machine learning, but one of their main focuses is reinforcement learning. So this is ah, This is pretty much what are Laff chef is in an offshore. Ehsan Kamalinejad: Um, again, there's a lot to talk about here, so I'm not gonna go into the detail of that. If you guys are interested, we can talk about them later. Also, I'm gonna do a little bit of marketing for my own course here. So I'm create, I'm creating a course in Rlf Chef that it's it's free on YouTube, you can go ahead and take a look at it. I will explain pretty much all of the pieces here. And in fact, that course, is based on this diagram from Claus Allega. 00:15:00 Ehsan Kamalinejad: Ahh, so that's that before I continue. Let me show you in real time. How the supervised fine-tuning ah changes the behavior of the model. If you have like a base model, let's say that it's a GPT or a llama or opt from Facebook or I don't know. There are many of them, you saw it free. And the idea is that, how does the fine-tuning change the behavior of the model and let me in fact Ah Show you a quick demo of that before I continue to the last few slides. Ehsan Kamalinejad: Okay. Ehsan Kamalinejad: Okay. Ah, can you guys see this? Ehsan Kamalinejad: Okay. Ah, Amir Feizpour: It could be a bit. Well, a lot bigger though. Ehsan Kamalinejad: all right, let me How's that now? Amir Feizpour: But thank you. Ehsan Kamalinejad: Okay, so this is this is the This is a fine-tuning that I did myself with some of the data that me and my colleague have collected. It's a extension on top of alpaca data from Stanford. Ehsan Kamalinejad: What we do is that we get the llama model. This one is one of the smaller models. Llamas, seven billion parameter from Facebook and then we fine, tune it. This is supervised fine. Tuning we are in the process of completing the Arla chest and but ah, this one is the first stage, which is the supervised fine tuning. These are the kind of these are a bunch of questions that I have here just as a way to show how the model changes behavior. After fine. Tuning, feel free to shoot a questions if you have for the model, I can do it in the live demo here. But let's take a look at a few examples of running it in. Ehsan Kamalinejad: A live. So this is the base model. And the question is, what are the differences between dogs and cats? And the answer is dogs and cats are both domestic animals. Good. They are both for leg. That's good. They both have fur. Sure, ah you see this is not. First of all, there's a bunch of other things here that I like Ah, This is next word prediction, right? It has behavior. That is not necessarily desirable. Ah, secondly, I'm more importantly It didn't really answer my question that I asked What are the differences between dogs and cats? Ehsan Kamalinejad: It good. That it came up with some of the similarities, but it didn't really answer my question. Now, this is the fine-tune version of the same model fine-tuned on a modest size, fine-tuning data set. And then it says, Dogs and cats or both popular pets, but they have distinct personalities and behavior, Dogs are often more energetic. And outgoing war, while cats are more independent and loose dogs. Typically need more exercise and attention than cats while cats are blah blah. Ehsan Kamalinejad: So you can see that it's the same model but when we do fine-tuning it behaves the way that we expect. And that's the that's the main source of the huge success of Chat GPT and Gpt45 because they found a way to tune these models in a format. That is really usable. Now let's look take a look at a few more examples and then I'm gonna not I'm not gonna bore you with more. Ah, let's see. And generally, you can see that the fine-tune model works a lot better. I want to. Oh, but that's this boiler. I have a good example but I'm afraid that it might be a spoiler. So let me stop. Let me just do this one. So here, I'm asking give me a list of three movies about zombies. Ehsan Kamalinejad: Okay, so it does something. I don't know exactly what it's doing. Ah, Ah yeah. There's like at least they I don't know what it's doing but if you look at the fine-tune model, It does exactly what I asked it to nikov the living dead blah blah, dawn of the dead. They are the dead and so on and so forth. So let me not go over the most interesting example, because this one is a spoil. It has a spoiler in it. So if you haven't watched last of us or maybe just cover your ears, For a second. So, 00:20:00 Ehsan Kamalinejad: The first. the question is explain why dual ah, While Joel allowed Ely to be killed at the end. That. Oh my God. Don't listen. Don't listen. Ehsan Kamalinejad: at the end of the last of us, And this is the problem that you can see ah well still Dex define tune exam. Answer is better but both of the model are doing hallucinations. Before the fine tuning and after fine tuning, I wanted to show that it doesn't solve all of the problems for us. Okay, let me go back to the to the slides. I'm hoping that I got it. Okay. Amir Feizpour: Yes. And why while you're doing that, can I ask you a question? So, when you talk about fine-tuning,… Ehsan Kamalinejad: Yes. Amir Feizpour: I assume you are referring to fine tuning on a question. Answer per data set versus auto. Tuning. Ehsan Kamalinejad: Ah, well. Ah, the question and answering setup can be formulated in the autoregressive. Also. So there are ways to turn one of them to another one and both of them work, okay. And at least right now, I don't think it's clear for anyone which one works better. And in my experience, they work at the same level at least to the data sets that I okay. Amir Feizpour: Also. Quote, from the chat. You said the modest size data set was used for fine-tuning. Can you quantify it? Ehsan Kamalinejad: Yeah. Yeah, in fact and I think here I will go over one of the frog. Oh, it's the nexus line. I will go over that in a second. So, So supervised, fine-tuning versus RL HS. What are the pros and cons of each of that approaches? Ehsan Kamalinejad: So think about that, you have a query and you have to write down an answer for that query as a labeler, It's a much harder task compared to say that you have a query with two answers and you say that, I like this one, I dislike this one. Just compare the level of the, the work and effort that is needed for these things. Ehsan Kamalinejad: So so in that sense, our alleged is preferred. However, for supervised poin tuning, you can just crawl a lot of data from the Internet. There are sites, for example, stack overflow, that there are questions and then the responses to those questions. So that is a lot easier to collect data for supervised fine-tuning versus architects. Um, So another weakness of our relative is that the data is collected online, meaning that the model should run the query and create some responses and then, the responses should be should be labeled while in the supervised fine-tuning, there's no modeling involved in the data creation, so it's offline data collection, which is, which is better. I mean easier. Ehsan Kamalinejad: But another good thing. This one is probably one of the most important things is ML in general, is the problem of catastrophic. For getting whenever you do fine-tuning Whenever you do transfer learning, this phenomenal happen that your model forgets, some of the knowledge from the past. Our hsf seems to suffer a lot less compared to supervised fine-tuning. This is a, This is a huge important thing about our religion. Last thing that I didn't, I forgot to mention here, I mean, I kind of mentioned it. RLF is a lot more complex setup compared to supervised finding Okay. So here are some important questions. And I think the first question is about the, how much data you need for supervised fine-tuning or our chef. So, if you have a base model, Ehsan Kamalinejad: How much data you? You need in order for this model to converse, to some good good answers in your specific domain. this is, Open question. I don't think. Research community in machine. Learning has come and conversed to a good answer yet. However, we have some empirical results, the empirical results show that you need in the order of tens of thousands of samples to UF. Good job of fine-tuning. Your language model, ah, with the with, with either supervised fine-tuning or are the chefs. So for example, the example that I showed you, which was from Amazon that was won over the, I think it was 60,000 or so samples. 00:25:00 Ehsan Kamalinejad: Um so a few tens of thousands of sample note that this is really, really a small compared to general type of data set that you require to train deep neural network models. So in the age of big data having like hundred thousand samples is not is not that big Ah, okay. Ehsan Kamalinejad: The other question is, how much improvement you will get on top of your base model when you do fine-tuning. This is also a question that is very hard to answer, mainly because they're right now, there is no good measure. Agreed upon measure that you can use to measure the quality of these models benchmarks such as Super Glue and those other benchmarks in traditional benchmarks. In language, These models are just saturating. Those those benchmarks. So they work very well for for typical questions in the benchmark. So it they're no longer very applicable. Ahm meaning that we need a better and better metrics to measure how good these models are. Ah, the next question is the scaling laws. Ehsan Kamalinejad: It's very important. So you, you want to know if you have limited compute, and you want to apply that limited compute to your training. There are the places that you want to assign those limited compute too. Do you want to go and get more tokens basically larger data set or you want to go with the larger model? And what is the trade-off in each of these things? So the Chinchilla paper from Deepmind came a few years ago and that was the, that was one of the good papers that came. Ah, came out to answer the question of a scaling laws and there are a few more recent ones. But these questions, these scaling laws so far has been all about pre-training. Now the question of fine-tuning is still vital. ah, Ehsan Kamalinejad: Finally, how effective our methods such as pest in fine-tuning. If you want to do lower or some other type of parameter efficient, fine-tuning, how effective they are. And if they are effective, how do they compare? So these are also very important questions. Some of these questions are being answered right now by the research community. So we have some partial answers for some of these, but they are not completely fully understood yet. So with that, I'm gonna stop here and see if there is any question. I saw that there were a bunch of questions in the in the text but maybe I'm here help me with the Amir Feizpour: Yeah. Yep. So let's start from here. So when you're talking about fine-tuning are we changing all the parameters or only a specific set of parameters? Ehsan Kamalinejad: Right. So that's again, that's one of the questions. In fact, it's related to the ahm like, the methods such as Path. So Path is one method that is. Try to do parameter efficient fine-tuning in the simplest setup. You just free some of the layers or some of the weights of the model and then fine tune the remainder of them all. There are all some other methods such as Laura that you actually add, you completely freeze the model and you add new new additional fully connected layers that they do the work of fine tuning and there are also other variants of this. So there are a bunch of different variants and doing that. Each of them has strength and weaknesses. So I cannot say that which of them is like a complete winner. At this point, I don't think anyone knows. Amir Feizpour: All right, so another question. And I'm gonna expand it a bit since. The general theme of what we're trying to do in these workshops is to tell people what they can do at home. So already, our lhf SFE, etc. most of the time when we were talking to them, I mean, 00:30:00 Ehsan Kamalinejad: As it just me or… Amir Feizpour: Of figuring out. Ehsan Kamalinejad: I think Amir you are cutting out. Ehsan Kamalinejad: I think it's not just me because I can see some of the other faces moving, so probably, it's Nikhil Varghese: Yeah, not just you. Ehsan Kamalinejad: Okay. So maybe while Amir is gonna come back with me, go over. Ehsan Kamalinejad: or maybe go ahead guys and because The moderator. And that's your, you just go ahead and ask your questions. Maryam Farooq: If you want to just take questions from the chat, feel free if there's something that you want to elaborate on go and

Jump it. Ehsan Kamalinejad: It seems that Amir's coming back, but okay. We have a little bit of technical difficulty here. Amir Feizpour: Hello, sorry. I'm back. Ehsan Kamalinejad: I am here. We didn't get your question. You were disconnected. Amir Feizpour: Okay. The matrix is a bit funky here today, so, I was Ehsan Kamalinejad: Yep. You were to saying something about our relative,… Amir Feizpour: I forget what I was asking, so yeah. Right. So so the General Right? Ehsan Kamalinejad: I don't know and like they're smaller scale. Amir Feizpour: So the general theme of what you're doing in these workshops is What can I do at home? so, you know, one question in the chat that I was trying to expand on is What what can I do with these at home? Right? Like Are there frameworks there? Where I can do rihf on a smaller models that I can run on small? You know, models that I can run in kolab? It's so if you could give, you know, some practical tips, for how these techniques that you talked about can be used. Ehsan Kamalinejad: Yeah. Ehsan Kamalinejad: Yeah. Amir Feizpour: Are all. Machine. Ehsan Kamalinejad: So it's a good question and for a lot of the researchers, especially junior people who are just learning and they want to get some experiments going, of course, it's a challenge to get like a machine with eight GPUs 10 GPUs and like it train at large models. So, but the fortunate thing is that some of these methods allowed the the models to, to be shrink like not actually shooting, but they freeze. A lot of the parameters of the model and then you can train them on. Like at home, you still need a GPU. But for example, we did GPU with the colon GPU, you should be able to run like a 20 gigabyte, 20 year, billion parameter model there is, in fact, I will send you ahmir a link about one of these documents from hugging face that shows you, how Ehsan Kamalinejad: Exactly, you can do that and train our lhf. With a limited compute. I believe that they were actually running it on at Colette machine. So it's possible to do, it's a lot of slower and you are limited to a kind of like a smaller models. You cannot go all the way to a hundred billion parameters or things like that. But for research experiments, I think that should be enough. So it's possible to do that. Of course it's not easy to set and set up Dad you should be you should buckle up for like very hard complicated setup, but it's possible. Amir Feizpour: Yeah. so, I want to qualify a little what I was saying by, you know, doing something at home. So you know, between what is available on, open AI. And what I can do at home for side projects, there is definitely things that people are trying to do in you know, companies They're trying to put models like this in production and you know for a lot of us it is important to make sure you're not sending all the IP to open AI,… Ehsan Kamalinejad: Right. Sure. Amir Feizpour: even though contractually they say they won use it. But who's to say it was, you know, a good person here. So, What are some of tips and tricks around using these things? You talked about for models that are intended to go to production. So these are not models that I can ask the chicken come first or… Ehsan Kamalinejad: All right. Amir Feizpour: egg and have time to sit down for ten minutes to wait, right? Ehsan Kamalinejad: Yeah yeah. So the fortunately the community has put together some forces in order to create some open source, both models and data sets. So probably some of you are already familiar with open assistant effort. That one is a exactly, the an effort to do our elitive eventually. They are at this point similar to what I showed you are doing super wise, fine-tuning, mainly, because supervised fine-tuning, the setup is a lot easier to do, but they're collecting data for all the chef and that's gonna be open source data set and you can apply it Note that these methods all model agnostic. So you can apply to any model. So, depending on what kind of compute you have it at your company, or you're at home, you will be able to apply those kind of a kind of methods. And those kind of data sets and supervise fine-tuning or are the chef 00:35:00 Ehsan Kamalinejad: Top of your model. The models. For a single machine. Usually the best in my experience, the best models are currently open source. Not open source, it's a license for research is Lama, Llama from Facebook, is very good model, they have good performance and slightly worse than that, not by big margin, but it's likely worse than that. Is elusive. AH, models PCR. Ehsan Kamalinejad: Ahh, I will send a link to those. Those are in my experiments. They have been the second best and they have very very sizes. They start all the way from like a 1 billion parameter that you can easily run in your laptop all the way to, I think 60 billion which is a sizeable sizable model. So in Europe at your company, you probably want to spin up a 60 billion parameter model and then you do the initial fine-tuning with open source data and then release it to your workers and get feedback and fine-tune. It more note that the RL surf is, is actually like kind of live kind of setup. So it's continual setup. So if you have a startup and you're interested to Ehsan Kamalinejad: Do a lot of these computes somewhere. I might be able to help you with with AWS. So, so that's also an alternate. Amir Feizpour: Erase. The next question, I do have a point of view about it but I want to hear what you think first and then maybe we can discuss it for a minute. They're asking other tools to collect data specifically for prompt and answer pers to train our agent. Ehsan Kamalinejad: You mean ah, like the something like a label box like a, right, like a labeling tool,… Amir Feizpour: For example. Ehsan Kamalinejad: there are a bunch of labeling tools. Some of them are paid the best one I think is from selai. If you search a scale, ai Allr chef, you will find the, ah, The the basically application and tooling that they are providing. It's paid, it's pretty cheap. So if you are have a company that's a good start and they they're work is super, super clean. Ehsan Kamalinejad: But then if you want wanted to be completely free, there are a few things open source that I've seen the quality of the of the setup. And the interaction is not that great for the open source so far, unfortunately, Open Assistant itself, for example, has a has a labeling, AH, application to there's one that some of my colleagues at AWS are working on and that's gonna be also free. Then it's published I think the quality of that is going to be slightly better, still not at the level of a scale AI so several different tiers. If you are willing to spend a little bit, go with a scale AI if you want to make it free and it's immediate open assistant if you want to do a little bit better. But three, wait for AWS is gonna come up with something. Amir Feizpour: Yeah. so, This is what I wanted to talk, you know, sort of an idea that I want to bounce off of you. One of the interesting things that we saw people do. With large language models is essentially data generation like synthetic data generation. So do you see a way? Ehsan Kamalinejad: Sure. Amir Feizpour: That if you were to try to train your own internal model and you know, do instructs instruction fine-tuning with. All right, our lhf and similar methods. Is there a way to just collect enough data? Throw it at, you know, open AI API, get a bunch of training set that you can use on your small model. And you know I trade on that a bunch of times and as you said like once this is in production, it's going to be your data machine that can you know, increasingly get better. 00:40:00 Ehsan Kamalinejad: Yeah, so that's a very good question. And the answer of that is, is complete It's yes, you can do that. However, there is a however, and however is about licensing and legal issues. Open AI clearly estates that you cannot there, use their API and also the, the interactive application. For training your own models. Ah, meaning that well feel free, if you feel that. Ehsan Kamalinejad: You want to do it, you can do it. And it's essentially a student, a teacher student set up, right? The teacher is gonna be in simple words. The teacher is gonna be gpt4 or chat GPT, and the student is gonna be your model You basically distilling the knowledge from their model, to your model to your model, it's completely possible. I have done a version of it just as an experiment. In fact, it works very well, but then the question of like maybe if you are keeping it to your small company, maybe it's okay but then you cannot really release that and get, you know. Ah yeah. That is a challenge over there. Amir Feizpour: It. Yeah, as long as you're unimportant, I don't think anybody's gonna come after you. But if you become an important and… Ehsan Kamalinejad: Exactly, exactly. Amir Feizpour: you're in trouble, most probably Amir Feizpour: Okay. Oh, there's one more. It feels fine. Tuning has been made tougher and has barriers to entry. What are your thoughts on fine-tuning paradigm? And how it could be accessible to everyone. I think we've talked about this. Ehsan Kamalinejad: Yeah. Amir Feizpour: Are verdict. Like what are the takeaways for people? And this topic of what they can do with this in their settings? Ehsan Kamalinejad: Yeah, I think. That's the that's the principle that I follow and it has worked very well for me over years. And I think specially recently and probably that's one of the good methods. I'm not saying that that's the only one but just Forget about the noise and stick to the principles. Ah, there are methods that you can see right away that these methods are not very specific type of models. And methods for example, applying RL to fine-tune a model. It's in fact, a very generic method It can be applied to any man and model anywhere. In fact, it they started. The RL tip is started not in language domain but it is started in robotics Ehsan Kamalinejad: So this is a very general method, so I think that that's gonna, that's gonna remain to be one of the players, but there are some other for example, fine-tuning methods, supervised fine-tuning that came up. It came out recently. So for example, soft prompts ah, prompt tuning and like the bunch of other things that for example, the fees like some of the tokens for the prompts and do some, these kind of things, I think they are super specific. That we will find ways that they, that are more general and they work a lot better in like soon. So don't really marry to any of the methods unless, you know, that it's like the principal is there. Ehsan Kamalinejad: And right now for fine tuning unfortunately, the best most principle approach is our LA chief and just the bang line, baseline. So supervised fine tuning. The baseline supervised fine tune has a lot of problems. Such as a cathetic forgetting. Our electric is a lot better but then it it's A The engineering side of it is complex. So on the theory side, it's not that complex but they engineering this company. So There's a little bit up trailer. Amir Feizpour: so, that brought up another point, which is A lot of us forget that ultimately these are machine learning models and Data has always been the goal and it still is the gold, you know. Not. None of these algorithms and frameworks are particularly defensible, but data is a mode… Ehsan Kamalinejad: Exactly. Amir Feizpour: because as we've been talking about it, It is hard to get the right data that you can train these models on. So that continues to be true, even though large language models are powerful and part of reason that they are powerful, is that? Well, they got trained on data that most other people don't have. So last thing before we sign off from the session, I know, AWS serves cohere models. do do they serve any other models, or You foresee,… 00:45:00 Ehsan Kamalinejad: Yeah, they have a new initiative that they and,… Amir Feizpour: I mean, Ehsan Kamalinejad: of course, there are a bunch of things that I cannot disclose. These are like internal Amazon stuff, but what they already have announced is the bedrock. Ehsan Kamalinejad: Initiatives that they are creating a bunch of different foundation and foundational models and they are collaborating with some of the some of the other labs in order to make the open source models, a little bit like widely more available. And for example, cohere is one of them that is working with AWS and TROPIC is another one and stability AI. In fact, I'm talking with both entropic and stability from Awsi. So there are collaboration going on and the, you know, from AWS perspective they don't they they usually don't make money out of the models. So they heard a model to be open source, They make money out of compute. So for them it makes sense to just like publish them data, the model, everything.