Speaker 1:

Okay, it is recording now. So first, I'd like to ask you some questions about your background. So the background questions can be answered briefly so that we can save more time for the other questions. So first, could you briefly describe your role in your team?

Speaker 2:

I work for deepset.ai to start up from Berlin. And we work on a haystack platform, which is an NLP search platform on top of hugging face. My role is a software engineer at the core team.1

Speaker 1:

Okay. So can you tell me a recent time when you use a pre-trained model from an external model hub?

Speaker 2:

From hugging face, for example?

Speaker 1:

Yeah.

Speaker 2:

I use them on daily basis, essentially.

Speaker 1:

Okay. So how do you use the pre-trained models?

Speaker 2:

I usually just fetch them with, from pre-trained in hugging face API.2

Speaker 1:

So do you usually retrain the model or fine tune the model?

Speaker 2:

Depends on the case. Most of the time, we just use the pre-trained fine tune model. Sometimes, I work on a making of a specific fine tuned model. For example, I work on a long form question answering where I had to make a sequence to sequence model that was fine tuned on custom data.3

Speaker 1:

Okay. So have you used any model as a backbone?

Speaker 2:

What do you mean by that?

Speaker 1:

I mean, like, use existing ways and then add some extended layers.

Speaker 2:

Ah. I'm familiar, but I have not done it recently.4

Speaker 1:

Okay. Okay, cool. So the next set of questions are about, how do you select Pre-trained Neural Network? So we're trying to understand the process that software engineers follow as they decide which Pre-trained Neural Network to reuse in their project. I think I show some definitions here. Here are definitions for the registries and attributes. And here for the attributes, we mean, the metrics used to measure the registries like, traditional attributes including the quality maintenance popularity and some machinery model attributes, which we proposed here in this project and are asked some questions about this three attributes later.

Speaker 2:

Actually, I don't do it that way. I first select it by architecture. So in hugging face, you have a top up button for, let's say sequence to sequence architecture. And then I look at that what is the latest, newest, the best model. And then I select it from there, using this approach.

Speaker 1:

Okay. So can you summarize a general decision making process which you follow?

Speaker 2:

So the general process is, I first think of an architecture. Do I need an encoder-decoder? Or just an encoder or just a decoder? So I already know the use case and what approximate architecture could be meeting the objective of my use case. So then I just go to, let's say, sequence to sequence, encoder-decoder, and then I pick a model from there.

Speaker 1:

Okay.

Speaker 2:

That is a kind of general framework. Admittedly, it requires some background knowledge about NLP models in the architectures, in order to start this search this way. But I found it most intuitive, at least for me.5

Speaker 1:

Okay. So, where do you usually choose pre-trained models? From a model hub or the pre-trained models from the open source project in GitHub?

Speaker 2:

I just go to huggingface.co, and there I pick models. And from there, I'm just doing it now. So for example, okay, so. Oh, this is called tasks. Sorry.

Speaker 1:

Okay.

Speaker 2:

So, they have a task list where... Well, this could be, let's say summarization, because I know that this kind of a task is sequence to sequence. Then, I would look at these summarization models or another task. Or is this, I think it is called text to text generation.

Speaker 1:

Okay.

Speaker 2:

So, okay. Basically, if you go to the hugging face and then models, then by a particular task, you can select what kind of... You can narrow down the search for a particular model.

Speaker 1:

Okay.

Speaker 2:

That is how I do it.6

Speaker 1:

Okay. So when you select the model, so which one do you care more about, the model performance or the model architecture?

Speaker 2:

I look at both. I usually look at the... I try to read a bit of a background, what is the state of the art in this. And then I see what kind of models are popular or the newest. That is the general approach I use.7

Speaker 1:

Okay. Do you think the pre-trained models available in the model registries can accurately describe their behavior?

Speaker 2:

Well, they have a really good cards now, depending on the author of the model. So some users... Sorry, model authors provide cards on hugging face, which not only show how this model is used, but they also give references to performance. They give references for the research paper. So I found these model cards to be really useful.

Speaker 1:

Okay.

Speaker 2:

I'll be... It is important that the author prepares these model cards.8

Speaker 1:

Okay. To what extent do the discrepancies of performance metrics affect your decision making? Here I mean, the actual performance of a model can be different from the claimed performance in the model card. So to what extent was that affect your decision making?

Speaker 2:

Minimal.

Speaker 1:

Okay.

Speaker 2:

I'll probably try the model myself and then see what kind of results do I get.9

Speaker 1:

Does that mean, you were retrial or revaluate the models?

Speaker 2:

I don't, but I should. I've heard that there is some talk now in hugging face where they plan to automatically evaluate these models with this new evaluate library that they have.

Speaker 1:

Yeah.

Speaker 2:

Which would be the independent measurement of performance, which I would definitely look at.

Speaker 1:

Okay. But you are not do that by yourself, right?

Speaker 2:

I don't. At the moment, I don't.10

Speaker 1:

Okay. So to what extent does the robustness of the models affect your decision?

Speaker 2:

I have actually thought about that. In term... What do you mean by robustness, exactly?

Speaker 1:

Whether the model can have good performance in terms of whether it has a server attack, different scenarios.

Speaker 2:

Like, even out of the sample data kind of a thing.

Speaker 1:

Yes.

Speaker 2:

Haven't thought about that, but I take into account the data distribution shifting. In terms of the data that the model was trained on and the data that I'm dealing with. I try to understand if the model will perform worse on the data that I have.11

Speaker 1:

Okay. So to what extent do you think the explainability of the model will affect your decision?

Speaker 2:

Yeah. Well, it would be great to see some of this, but I have not. I mean, I would love to see some kind of explainability about the... In the model card or how to invoke it or measure it, but I haven't seen anything like that so far.12

Speaker 1:

Okay. Okay. So, how frequently do you retrain the model?

Speaker 2:

Like, pre... Sorry, fine tune them, you mean? Or,

Speaker 1:

I mean, retrain the model from scratch rather than fine tune.

Speaker 2:

Oh, from scratch. I've done birth training a couple of times from scratch, but not recently.

Speaker 1:

Okay. Yeah. The next question is, how frequent do you fine tune the models?

Speaker 2:

Yeah. We do that maybe, let's say, once a month.

Speaker 1:

Okay. So are those like fine tuning the same model or like once a month in different models?

Speaker 2:

Different model, depends on the task. Right now, I'm working on some specific task that will require me to fine tune a certain model. And then, I'm still generating data for it, but I mean, I'll fine tune it within a week or two.13

Speaker 1:

Okay. Do you think the lack of trainability or finetuneability is a problem while using a pre-trained model from the model registries?

Speaker 2:

I don't think it is a problem.14

Speaker 1:

Okay. So what other challenges do you have when selecting a pre-trained model from a model registries?

Speaker 2:

Yeah, it would be great to see if I... How when you buy stuff on Amazon and you select a book and then it gives you very similar models, that would be great to see something like this in hugging face hub or somewhere, on some registry.

Speaker 1:

So, it is more like a model recommendation.

Speaker 2:

Exactly. Model recommendation. Yeah.15

Speaker 1:

Okay. Okay. Sounds good. We'll move on to the next set of questions, which is about depending software attributes. So, we would like to learn about what sort of information is useful for engineers who reuse the pre-trained models. So I have some definitions here. So, MPM defines the following attributes for the JavaScript packages. And you can spend some time looking at the first sentence or first paragraph of each HBS here. And let me know when you are ready.

Speaker 2:

Ah-ha. Okay, cool.

Okay. I intuitively search information about all three of these. So what is your question then?

Speaker 1:

So, what do you think would best help your team select a pre-trained from the model registries?

Speaker 2:

Most likely, popularity.

Speaker 1:

Okay.

Speaker 2:

That is a cheap filter. Then, after that, I would look quality, which would be presumably more information, the card about everything, how it was trained? How to use it? When was it last updated? What is the stable version? And then I would look at the maintenance, last.16

Speaker 1:

Okay.

So then, we define the following three attributes. So I will ask several questions about each of these attributes now. So we can start from the prominence here. For the prominence, we defined as a measure of model lineage or traceability. So here are some examples, like the link to the paper, whether it is something relevant to a research or commercial group, whether it has a GitHub page, data set or architecture information in the web model card. So can you think about a time when you met some provenance problems when you reuse the pre-trained models before? Can you tell me the challenges you had?

Speaker 2:

Well, the challenge is that the authors did not fill out the model card, it is up to them. I made some my own models, fine tuned them for long form question answering. Nobody chased me to fill out this information, I made it myself to the extent possible. But I've seen models where users do not fill out this information. So yes, it is annoying not to know anything about how it was trained? Where is the data that was trained? And all of these things that you mentioned.17

Speaker 1:

Yeah. So what do you think would be useful to know beforehand in order to solve these problems?

Speaker 2:

What do you mean? Well, I would love to know all of these that you mentioned in provenance category.

Speaker 1:

Is there anything else?

Speaker 2:

It would be great to have a link to a GitHub page, how to recreate the data set? Sorry, the model, how to fine tune the model yourself, if you want to do it?18

Speaker 1:

Okay. Okay. Then we'll talk about the reproducibility. So here, I would define as the ability of developing practitioner to produce the same accuracy and training or evaluation time from a Pre-trained Neural Network as defining the paper source code or as the group. So here are some examples about the reproducibility, like, whether the model card contains hyper parameters, the configuration, whether it provides any dock image like, demo for the model. So can think about time when you met any reproducibility problems when you used the pre-trained models before?

Speaker 2:

Hmm. Not that I can recall explicitly, but I'm sure that there were some occasions. Yeah, it would be great to have a demo right there on the model card where you can play or a notebook demo. Yeah. Inference level. Yeah. That is really important. So you can immediately start to play with the model.19

Speaker 1:

Okay. So except for these bullet points here, do you think there are any other things that can affect the reproducibility for reusing the model?

Speaker 2:

No, man. You listed many of them. I would've thought of one or two, three of them all.20

Speaker 1:

Okay. Okay. Then we'll talk about the probability. So here we define as the ease ways, which engineer can take a Pre-train Neural Network and reuse it in another environment, software project or anything else. So here are some examples, like the Android consumption, whether it can be used in CPU, GPU or TPU, which different framework you can use. So can you think about time while you have any the probability problems before?

Speaker 2:

Yeah, I can think of a time. So for example, now I'm trying to run these large... Sorry, large language models. And not all the time is it explained how to start them because most of them require multi GPU setups. But, you dig around and you figure out that there is this auto configuration in the new hugging face, the latest hugging face release, which splits the model on multiple GPUs, shards it automatically for you, and then you can use it. So, yeah, I'm sure other people face this issue, but a bit of a digging, typically solves this problem, but I can see that this would be for me, the biggest issue that people do not address. For example, you take these T zero PP model on the hugging face. Unless you know how to do this, I'm not sure you will be able to even start a model. So,

Speaker 1:

Okay.

Speaker 2:

I can see this being an issue for large language models in the near future21.

Speaker 1:

Okay. So except for this, do you think there are anything can be useful to know beforehand in order to solve the probability problems?

Speaker 2:

Yeah. I can't think of anything else I just used [inaudible 00:17:55]. So I'm just focused on that framework. Yeah.22

Speaker 1:

So, have you had any issues about the latency of the models before?

Speaker 2:

No, but you account for it while working with a model. And yeah, you figure it out what it is and from experience, basically.23

Speaker 1:

Okay. So the last question here for this part is, except for these three attributes, do you think there are any other attributes could be helpful for the pre-trained model we're using?

Speaker 2:

Mm-hm. I think this is going to be an issue for large language models because they'll require more setup. So unless you are really proficient and have the background knowledge, I can see how large language models are going to be problematic to run.

Speaker 1:

Okay.

Speaker 2:

So I'm not sure how you can you probably can figure out from this answer, something from all three of these categories.

Speaker 1:

Okay. So you mean, there is no Azure attributes you can propose here?

Speaker 2:

I cannot propose attributes. But... I cannot think of them on top of my head, but I can see that the issue is going to be how to run these 11 billion parameters.

Speaker 1:

Okay.

Speaker 2:

So, what you need to do? How to prepare? And... So, I guess it falls under portability.

Speaker 1:

Yeah. Yeah. I think so.

Speaker 2:

Yeah.24

Speaker 1:

Okay. So the last set of questions is about the pre-trained model trustworthiness. So we are trying to understand how the Pre-trained Neural Network's shortcomings affect engineer's ability to rely on and reuse them. So the first question here is which aspects of the pretender network do you assume a trustworthy in the model registries?

Speaker 2:

Well, usually, that it comes from reputable names in the industry.

Speaker 1:

Okay.

Speaker 2:

That is how I measure trustworthiness now.

Speaker 1:

Okay.

Speaker 2:

I know most of the players in the industry, even individuals, so that is the main kind of a decision making input that I use when it comes to this.25

Speaker 1:

Okay. So have you any discrepancies between the Clam Pushen models and the downloaded version, in terms of the accuracy, latency or architecture?

Speaker 2:

I haven't, but what I found is that not many people are talking about data distribution shifts. So I think this is something that should be mentioned. Like, that the model was trained on this specific data. If your data is a bit different, then your experience will likely degrade. Sorry, accuracy of the model.26

Speaker 1:

Okay. So to what extent do you think the discrepancies are acceptable? Like, in terms of the accuracy, latency and architecture.

Speaker 2:

Like... Hm. I haven't thought about that. I would just simply not use the model if the latency is huge. Usually, you take everything... What people say is actually true and try to find a bug on your end. And this is an exhaustive process sometimes.

Speaker 1:

Yeah.

Speaker 2:

So, that is something that I would like to highlight.

Speaker 1:

Okay.

Speaker 2:

So, the way I addressed it is that if there is a large discrepancy and I cannot close it from the claimed performance and model card, would usually contact the authors and say, "Listen, I'm getting this kind of a performance. What do you think is the cause?" So this is how I would address it.27

Speaker 1:

Okay. So the last question here is, do you think the discrepancies would have significant impacts in your product, let's say?

Speaker 2:

They would. This is one of the... How can I say? The most time consuming kind of discrepancies that usually shows up because when you run it on your hardware or the framework version, something pops up. It is like, a performance is not the same or the accuracy or... And then, you try to figure out what the hell is going on. So this takes a quite a bit of time. Okay. Yeah.28

Speaker 1:

Okay. All right. That is all of the questions for my interview. I will stop.

Speaker 2:

Oh, really?

Speaker 1:

Yeah. I will stop recording now.

**Annotations**

1 Subject is a Software engineer at deepset.ai working on a NLP search platform

2 Subject fetches PTNNs from huggingface's API daily

3 Most of the time subject does not fine tune models

4 subject has not "recently" used a model as a backbone

5 Subject selects architecture first, then sorts models on newest date

Seems like subjecs fall into 2 camps where architecture is vital or ignored

This subjects decicion making process requries knowledge on NLP model architectures

6 Subject uses hugging face, filters by "model" first then "task list"

7 Subject considers both architecture and performance

8 the subject seems to believe cards are "really good now" but awknolodges that it is author dependant

9 Discrepancies don't affect the subject because they test models themself

10 Subject does not retest the models to validate claims but awknoledges he probably should

Subject would prefer an indepant measurement were provided

11 Subject has not concidered robustness

Subject has concidered data distrobution shifting

12 Subject would like to see an imporvement in explainability

13 Subject has trained models from scratch "a couple of times"

Subject fine tunes a differnet model once a month

14 Subject does not believe trainability or finetunability is an issue for PTNNs

15 Recommended models would be useful

16 Popularity is #1

Quailty is #2

Maintanince is last

17 The fact that authors don't need to fill out model cards is a challange

Sometimes models or the data they are trained on are not well described

18 In additon to all the providence information given, subject would like to see a github link to recreate the data set and instruction on how to fine tune.

19 Subject can not recall reproducability issue

A notebook demo of a model would be nice

20 Subject can not think of additonal reprocudability issues

21 Subject gives an example of how instructions for runnning a model on multiple GPUs is not clear

22 Inaudible

23 Subject has not had latency issues

He accounts for them while working with the model

24 Large language models will require more setup and this could become problematic

This may be a portability issue

25 Subject trusts "reputable names in the industry"

26 Subject has not experinced discrpancies between claimed results and the downloaded models results

Subject is aware of the issue of data distrobution shifts

27 Subject has not conidered accepable discrepancy limits

If latency is too high - the model will not be used

If the discrpancy seems too big the subject would reach out to the authors

28 Discrepancies take time to troubleshoot so they have significant impacts