**Name:** Interview-10-20220803

**Created On:** 8/9/2022 4:49:34 PM

**Created By:** WJ

**Modified On:** 8/12/2022 6:50:25 PM

**Modified By:** WJ

**Size:** 11 KB

Speaker 1:

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

Speaker 2:

Basically, I was the one doing the coding engineering.

Speaker 1:

So what kind of differing models are you working on?

Speaker 2:

So basically transformer architectures only and pre-trained Neural Networks.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:

Yes. So from HuggingFace Hub only, I have used a lot of models, like BERT-based and [inaudible 00:00:59], et cetera. Long formal.

Speaker 1:

So what did you do for pre-trained model? How do you use that?

Speaker 2:

So basically I fine tune it on downstream tasks, and then use it for the evaluation. And also, basically my work was... so it is under review currently, but it was something about predicting the... so basically, understanding the dynamics of fine tuning and predicting the generalizing behavior of fine-tuned models.

Speaker 1:

Okay.

Speaker 2:

So we usually see that when you, for example, if you have a BERT model and you fine tune it, then you see that this is the... suppose you fine tune it on MLI. Then you get this much MLI validation score. But if you want to check on how does it perform out of domain, then you train it on the out of domain task. For example, there was a recent paper, so about enhanced data set. So basically it consists of... so for MLI, there is a premise and a hypothesis and you have to predict whether the premise entails the hypothesis or whether the hypothesis entails the premise or not. You have to predict that. But usually we see that models adopt shortcuts. They will try to match the words between premise and hypothesis instead of actually reasoning. So this, our work was to explore this behavior, like when does it happen? How can we remedy this?2

Speaker 1:

Okay. So have you ever retrained any model before?

Speaker 2:

Trained? Yes. Yes. Pre-trained, or fine tune?

Speaker 1:

Well, I mean retrain the... let's say you take the BERT model and you retrain the model from scratch and see what's the performance.

Speaker 2:

No, I haven't retrained it from scratch.

Speaker 1:

Okay. Okay.

Speaker 2:

But, so I think there are lots of retrained BERTs available already, in the Google repository.3

Speaker 1:

Okay.

Speaker 2:

If we want to.

Speaker 1:

Okay.

Speaker 2:

I use some from [inaudible 00:03:44].

Speaker 1:

So do you use any model as a backbone?

Speaker 2:

So I have used BERT, Roberta, long pharma. I have experienced at least three.4

Speaker 1:

Okay.

Speaker 2:

Using [inaudible 00:04:00].

Speaker 1:

All right. So the next set of questions is about the model selection. So we are trying to understand the process that engineers follow as they decide which pre-trained model you use in their projects. So can you think about the last time, why you choose a pre-trained model from a model hub? How did you choose it? Can you summarize your decision-making process?

Speaker 2:

So basically, the results from earlier papers were on BERT model only. So I chose the standard BERT model available on HuggingFace Hub. And then I fine-tuned it from there.5

Speaker 1:

So where do you select the pre-trained models? How did you determine you should choose a model from the model hub instead of the models in GitHub project?

Speaker 2:

Oh yes. Once I had to use models from GitHub project also, like a paper hat, they didn't have those models available on HuggingFace Hub, and they had only really released .zip files. So I had to take from them also, because I wanted to reproduce their results. But then later on I pushed those models onto HuggingFace Hub also because the integration, my entire code base, it was difficult to integrate it with .zip files or the local files. And because this year, also the code between different machines and develop.

Speaker 1:

So can I say you mean the HuggingFace models in HuggingFace is easier to use than the GitHub models?

Speaker 2:

I think it is more standard to use. So the code base I was working with already had a lot of interaction with HuggingFace Hub. So for me it was more standard to keep every model in the HuggingFace Hub only, because most of the models were already there.6

Speaker 1:

Okay. So when selecting the model, do you care more about the model's performance than the architecture?

Speaker 2:

I think both are related with each other actually. So also it depends on the requirements, which are [crosstalk 00:06:35].7

Speaker 1:

Okay. Okay. So do you think that pre-trained models available in model registries can accurately describe their behavior?

Speaker 2:

So what do you mean by behavior here?

Speaker 1:

So, I mean in the documentation or in HuggingFace, they use the model cars, they have the cams, some metrics. Do you think the metrics can accurately describe their actual performance?

Speaker 2:

The metrics that they have reported?

Speaker 1:

Yeah.

Speaker 2:

Yes. All the models that I fine-turned got approximately the same metrics as the ones they reported.

Speaker 1:

Okay.

Speaker 2:

But I think there was some difference in the time they had reported for fine tuning. My models are taking a bit longer to fine tune using the same script that they had provided. But later on I found what was the difference, but still, I think the time varies somewhat based on the architecture of the GPU that you are training.8

Speaker 1:

Okay. So to what extent do you think the discrepancies of performance metrics can affect your decision making? So here for the discrepancies, I mean, sometimes the actual performances can be different from the claimed performances in the model cars.

Speaker 2:

So I haven't actually encountered a case like that.

Speaker 1:

So you can imagine there is a model which has discrepancies of this kind of metrics. So would that affect your decision?

Speaker 2:

For using that model?

Speaker 1:

Yeah.

Speaker 2:

So definitely I think it'll impact.

Speaker 1:

Okay.

Speaker 2:

But it is impossible to know beforehand, without the metrics reported.9

Speaker 1:

So to what extent do you think the robustness of the models can affect your decision?

Speaker 2:

So my entire work was on differentiating the robust models from not so robust models. So basically, it impacted a lot.

Speaker 1:

So how do you find whether a model is robust from a model hub?

Speaker 2:

From a model hub? So I think one thing to try is... so there are some known aggregated data sets on data link, but what [inaudible 00:09:30] that you can do in the input and check the robustness. So you can first construct those or download those data sets and check the performance on those data sets, but still...

Speaker 1:

So you mean-

Speaker 2:

I think when the field progresses further, we will converge onto training methods that lead to more robust models. And then, so I don't think this is an issue that needs to be handled by the model register, but as a machine learning community, we need to develop better [inaudible 00:10:18].10

Speaker 1:

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

Speaker 2:

So I don't think I actually have had that kind of application.

Speaker 1:

Okay.

Speaker 2:

So basically, I think I did some explainability analysis for some models, but that also varies based on which explainability techniques you are using. So basically, the large models that are already posted on the modern registries, I don't think they are quite reliable methods to measure their explainability yet, or maybe I don't know of them.11

Speaker 1:

Okay. Okay. Next question is, how frequently do you fine tune the models?

Speaker 2:

Very frequently, like a hundred models per week, something like that.12

Speaker 1:

Okay. So do you think the lack of trainability or fine tune ability is a problem while using pre-trained model?

Speaker 2:

Lack of what? Sorry.

Speaker 1:

The lack of fine-tunability or trainability.

Speaker 2:

No, they are easily fine-tuneable. But I think as there were some data sets which were not available on the HuggingFace Hub or some tasks. I encountered some tasks, basically CRF had conditional random fees.

Speaker 1:

Yeah.

Speaker 2:

So instead of the usual thing they have in HuggingFace Hub models is usually for sequence classification or sequence to sequence. But so I think they should have CRF effects also.

Speaker 1:13

Okay. So is there any other challenges you faced before when selecting a pre-trained model from a model registry?

Speaker 2:

No, I don't think so. Selecting.14

Speaker 1:

Okay. Okay. So we'll move on to the next set of questions, which is about the different software attributes. So we want to learn what sort of information is useful to engineers who reuse the appreciation models. So here are some definitions from MPM. So these are the three traditional attributes, which is used by MPM, which is for the JavaScript packages. And can you take a look at the first paragraph of each attribute here and briefly look at your definitions here and let me know when you're ready.

Speaker 2:

And let you know what?

Speaker 1:

Let me know when you are ready.

Speaker 2:

Okay. Quality attributes are easy to calculate because they are self-contained easily. [inaudible 00:13:38].

(Silence).

Yes, so I'm done reading.

Speaker 1:

Okay. So for this three traditional attributes, what do you think would best help your team select an application model from the model registries?

Speaker 2:

Sorry, I didn't catch you.

Speaker 1:

Or can you give an order of these three attributes, which can help your team select the pre-trained models?

Speaker 2:

So I think these three are not really that important because when we are using models, I don't think it is suitable to classify under these three categories, because I think the major part was popularity, how widely that model has been adopted in the community already, because the fewer... when I want to use a model, I want to be sure that is because my main work is towards research. So I want to use the standard things only. So that is [inaudible 00:15:53] that I want, more general.

Speaker 1:

Yeah.

Speaker 2:

So popularity is one.

Speaker 1:

So what about the quality and maintenance? Do you care about these two metrics here?

Speaker 2:

So I don't think I have seen any model being maintained or updated that frequently. They usually, I put a model and that is done thing. They don't update it ever.

Speaker 1:

Okay.

Speaker 2:

But quality, the quality is important still. More important than maintain and such.15

Speaker 1:

Okay. Okay. So these are three different specific attributes we define here. And I will ask several questions about each of these attributes later. So first I will discuss the provenance, which we define as a measure of model lineage or traceability. And examples are whether the documentation has a link to the paper, whether it's from some famous research or commercial groups. So can you think about time when you met any provenance problem before?

Speaker 2:

When I didn't know the model lineage or traceability?

Speaker 1:

Yeah.

Speaker 2:

No, I don't think... so basically because I use the models from hub only because I know that I have the paper and then there is the link in the paper towards the models. So I follow that link and only then I use the models on the hub.

Speaker 1:

So does that mean if the model does not have the link to the paper, you will not look at these models?

Speaker 2:

No, no. The paper should have the link to the model, but not [crosstalk 00:17:50].

Speaker 1:

Oh, okay okay.

Speaker 2:

... otherwise, because they have to [inaudible 00:17:55] models, so.16

Speaker 1:

Okay. So what do you think would be helpful to know beforehand in order to solve these provenance problems?

Speaker 2:

So I think...

Speaker 1:

Or I say, what do you think the model registries can provide to help you solve the provenance problems?

Speaker 2:

So if we have supposed model and I want to know how this is trained or something. So one thing, yes. Basically I would like that to solve this problem that instead of just a model or... so the model, so they provide scripts. In HuggingFace, they provide scripts for fine turning models or pre-training models. And they log the models at continuous checkpoints. And together with that, I would like that they log the script also, so that I can identify what code exactly generated this model, like perhaps I usually integrate it with one [debriefs and biases 00:19:35].17

Speaker 1:

Okay.

Speaker 2:

Yeah.

Speaker 1:

Okay.

Speaker 2:

So.

Speaker 1:

Okay. Now we're going to talk about the reproducibility here. So here for reproducibility, we means the ability of a learning practitioner to produce the same accuracy and training or evaluation time from a pre-trained model as defined in the paper source code or the original group. Can you think about time when you met any reproducibility problems before?

Speaker 2:

Basically, sometimes with large models, I have that problem sometimes, because I don't have that necessary GPU or number even. But otherwise I don't think there was any problem with reproducibility.

Speaker 1:

So do you mean you cannot validate the result for the large models or you trust those models?

Speaker 2:

Yes. So I couldn't validate the results because I couldn't run the models.18

Speaker 1:

Okay. Okay. So in that case, what do you think would be useful to know beforehand in order to solve these kind of problems?

Speaker 2:

Basically, what can keep the logs of the evaluation, sample-wide evaluation, for example, on whichever data set they have provided the results for.

Speaker 1:

Okay. So what kind of logs do you mean here?

Speaker 2:

So for example, when I fine tune models, different kinds of architectures and different kinds of seeds, I fine tuned all these models on MNLI, then instead of just reporting the final scores, so when we evaluate a model, we actually evaluate it on all that data sets. So we can keep the label that the model predicted on each of these samples as an added that log to the model registry itself, that would be useful. I think HuggingFace Hub already has an evaluate functionality also.

Speaker 1:

Yeah.

Speaker 2:

So basically, so I think maybe they can add it automatically also if people request it, like sample or logs.

Speaker 1:

Okay.

Speaker 2:

Also this would be useful for error analysis, like which kinds of samples.19

Speaker 1:

Okay. So the last attribute here is portability. So we defined as, with which an engineer can take a pre-trained model and reuse it in another environment or software project. Can you think about time when you met any portability problems before?

Speaker 2:

I think basically, I was trying to use [Flax 00:23:43] and my code base was in Jackson Flax. And so HuggingFace doesn't actually have all the models in Flax right now. So I had to switch to ply touch for some part of my call, and then change back to Flax. That was sort of difficult. So I think it'll be better if HuggingFace can get all the models in Flax and Jacks also.20

Speaker 1:

Okay. So is there anything you think would be helpful to know beforehand for the portability problems?

Speaker 2:

I don't think especially.21

Speaker 1:

Okay. So the last question for this part is, except for these three attributes, do you think there are any other attributes that would be helpful for the reusing of pre-trained model from the model registries?

Speaker 2:

Can you give an example on what lines are you thinking?

Speaker 1:

So some examples are whether the model hub can provide the robustness explainability as part of attribute here. So something like that.

Speaker 2:

Okay. I don't think... one sec.

Speaker 1:

Okay.

(Silence).

Speaker 2:

I think for enhancing reproducibility only, we need that the code be committed alongside the models. Would that be...

Speaker 1:22

Okay. Yeah.

Speaker 2:

Most of the things.

Speaker 1:

Okay. So the last part here for our discussion is trustworthiness. So we are trying to understand how the pre-trained model shortcomings affect engineers' ability to rely on and reuse them. So, which aspects of the pre-trained model do you assume are trustworthy in the model registries?

Speaker 2:

So of which aspects are you talking about here?

Speaker 1:

So for example, whether is a model architecture reported in the model card match the paper. And also another example here is the performance with the pre-trained weight is correct.

Speaker 2:

So that is fine.

Speaker 1:

Okay. So you mean, you assume all of the pre-trained models are trustworthy?

Speaker 2:

Trustworthy, I don't think they are that trustworthy. When we do actual application, we need to add different kinds of checks and other things also. But-

Speaker 1:

So what kind of check do you mean here?

Speaker 2:

Application-specific checks on, so that basically we don't end up doing the wrong kinds of some things in our application.23

Speaker 1:

Okay. So have you found any discrepancies between the [inaudible 00:28:12] pre-trained models and the downloaded version? In terms of accuracy, latency, and architecture?

Speaker 2:

I think so basically, the papers usually report that we have this architecture and we fine tune this, or we pre-train this, but the work that I had done was revealing that the trustworthiness and robustness of the model basically can depend a lot on the random seed that you use for initialization. And initialization of top layers and the fine-tuning layers and pre-training layers and the...24

Speaker 1:

So-

Speaker 2:

[inaudible 00:29:06]. So if you would read the multiverse paper by Google, do you know? Okay. So that paper has explained some of these issues. And the recent art that I had done was it.

Speaker 1:

Okay.

Speaker 2:

But it is under [inaudible 00:29:29].

Speaker 1:

So how did you address these problems? Except for, okay, reading a paper which expanded well?

Speaker 2:

So I think we would need to retrain them on that in order to actually address these problems because these problems are present in the actual models available on hub only. So we have to... if the problems are actually in the rates of the model only, then we would have no choice but to retrain the model.

Speaker 1:

So here for retrained, you mean fine tune the model or retrain from scratch?

Speaker 2:

Sorry? Sometimes if the fine model is not robust, then you can try retraining from the pre-train [inaudible 00:30:34] point, but be a more robust, fine-tuning method.25

Speaker 1:

Okay. So the next question is, to what extent-

Speaker 2:

Also there is another interesting technique to improve the robustness of the models.

Speaker 1:

Yeah.

Speaker 2:

That are available on the hub. Basically, you take the models that are available and you average their weights. That tends to increase the out of the domain performance usually.

Speaker 1:

Okay. Okay. So does that mean you take several different models, which are fine tuned on different data sets and you take the average?

Speaker 2:

They are fine tuned on the same dataset.

Speaker 1:

Okay.

Speaker 2:

But different codes and different people. They have different seeds. So there is a paper on these weights. So [inaudible 00:31:33], no, that one is...

Speaker 1:

Okay. So are they have the same architecture, but they have different weights or different training configurations? Okay. Okay.

Speaker 2:

They have the same architecture, but different rates.

Speaker 1:

Okay. So the next question is, to what extent do you think the discrepancies of the models are acceptable if they exist in the model registries?

Speaker 2:

I think it depends on the application. Once I was trying with these models and my application had need for good collaboration, but there are some models on the hub that have accuracy is good, but calibration is very bad.

Speaker 1:

Okay. So in terms of accuracy, let's say, for example, do you think if the discrepancies of the accuracy is less than a certain percentage would be acceptable?

Speaker 2:

How much? So usually, all the applications are not on the standard dataset that have been reported in the paper, but on the custom data sets. So all that matters is how it performs on the custom dataset, not on the standard datasets.26

Speaker 1:

Okay. So the last question here is, do you think the discrepancies will have significant impacts on these pre-trained models?

Speaker 2:

So if the model is robust, then I don't think slight differences come from implementation. Also when you implement them, when you are trying to evaluate a model, and suppose once I tried it with 128 sequence length and another time I tried it with 512 sequence length, and the model has same parameters and I evaluate on the same dataset, but still, the performance can slightly differ because of the extra pack to concept I have added. So that much is, it depends for... slight differences can come from implementation to implementation.27

Speaker 1:

Okay. Okay. That's all for my questions. Thank you very much. I will stop recording now.

**Annotations**

1 Role:

- coding

- transformer architectures and PTNNs.

2 Reuse Scenario:

- From HF only.

- Fine tune it on downstream tasks. Then use it for evaluation.

- Understanding the dynamics of fine tuning and predicting the generalizing behavior of fine-tuned models.

- When does a kind of behavior happen? How can we remedy this?

3 Retrain: Never. There are lots of retrained BERT in the Google repo.

Fine-tune: Yes.

4 Backbone:

- I used at least three.

5 Decision making:

- Paper -> architecture (BERT) -> standard model on HF -> fine-tuned it from there

6 Where:

- GitHub project when they didn't have those modesl available on HF hub.

- Pushed models onto HF because the integration, also the code between different machines and development.

- HF is more standard to use.

7 Performance or architecture:

- Both are related with each other

- Depends on the requirements.

8 Accurate behavior:

- All the models that I fine-tuned got approximately the same metrics as they reported

- There was some difference in the time they had reported for fine tuning. My models are taking a bit longer to fine tune using the same script that they had provided.

- The time varies somewhat based on the architecture of the GPU that you are training.

9 Discrepancy:

- Definitely it'll impact

- It's impossible to know beforehand, without the metrics reported

10 Robustness:

- My entire work was on differentiating the robust models from not so robust models.

- It impacted a lot.

- When the filed progresses further, we will converfe onto training methods that lead to more robust models.

- This is not an issue that needs to be handled by the model register, but as a ML community, we need to develop better.

11 Explainability:

- That varies based on which explainability techniques you are using.

- The large models that are already posted on the modern registries, I don't think there are quite reliable methods to measure their explainability yet.

12 Fine tune: very frequently

Retrain: No

13 Fine-tunability/Trainability:

- No. They are easily fine-tunable.

- There were some datasets which were not available on the HF hub or some tasks.

- They should have CRF effects also.

14 Other challenges:

No.

15 Traditional attributes:

- These three are not really that important.

- Popularity > quality > maintenance

- I don't think it's suitable to classify under these three categories.

- Maintenance: I don't think I have seen any model being maintained or updated that frequently.

- Usually I put a model and that's done thing. They don't update it ever.

16 Provenance issue:

- No

17 Provenance help:

- Scripts for fine tuning models or pre-training models

- log the models at continuous ckpts

- together with that, log the script also. So I can identify what code exactly generated this model.

18 Reproducibiility issue:

- with large models, I don't have that necessary GPU or number even.

- I couldn't validate the models bacause I couldn't run the models.

19 Reproducibility help:

- Instead of just reporting the final scores, we evaluate it on all that datasets.

- We can keep the label that the model predicted on each of these samples as an added log to the model registry itself.

- This would be useful for error analysis, like which kinds of samples

20 Portability issue:

- HF doesn't have all the models in Flax/Jax right now.

21 Portability help:

- No especially

22 Other attributes:

- We need the code be committed alongside the models

23 Trustworthy?

- No. We need to add different kinds of checks and other things.

- Application-specific checks.

24 Discrepancy?

- The trustworthiness and robustness of the model basically can depend a lot on the random seed that you use for initialization.

- Initialization of top layers and the fine-tuning layers and pre-training layers

25 Address the discrepancy problem:

- We need to retrain them.

- Sometimes if the model is not robust, then you can try retraining from the pre-trained ckpt.

- Another technique to improve the robustness of the models: average the weights.

26 Acceptable discrepancy:

- Depends on the application.

- All that matters is how it performs on the custom dataset, not on the standard datasets.

27 Significant impacts:

- If the model is robust, I don't think slight differences come from implementation.

- Slight differences can come from implementation to implementation.