**Name:** Interview-07-20220719

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

**Created By:** WJ

**Modified On:** 8/10/2022 7:32:16 PM

**Modified By:** WJ

**Size:** 10 KB

Speaker 1:

All right. First I'd like to ask you some questions about your background. The background questions can be answered briefly, so we can save time for the other questions. First, could you briefly describe your role in your team?

Speaker 2:

Yeah, I'm a machine learning engineer that works together with a software engineer, and we're deploying a tool on Google Cloud platform. Yeah. For this, I usually try and fine tune our transfer-learn deep neural nets to teach them new downstream tasks in order to apply them to text. And then I would deploy them on Google cloud platform as microservices, by a fast API in Python. And then our software engineers integrates them into our front end.

We a team of two doing the technical work at our [inaudible 00:00:58].1

Speaker 1:

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

Speaker 2:

Yeah, I used one this morning. There was P5 that I used.

Speaker 1:

So what did you do with the model?

Speaker 2:

So this morning I was interested in figuring out how I can use keywords in order to generate a text from it. I was trying different approaches and I found a solution that someone used and he uploaded his model to a HuggingFace. And so I was downloading it from HuggingFace and ran some inference and experiments to see whether that's useful for us or not.2

Speaker 1:

So do you retrain or fine tune the models?

Speaker 2:

I sometimes do. Like sometimes I create artificial data sets or I use data sets that are out there to fine train models. Yes. Today I did not.3

Speaker 1:

So have you used any model as a backbone?

Speaker 2:

Yeah. Yes.4

Speaker 1:

Okay. Okay. The next set of questions is about the model selection. We are trying to understand the process that software engineers follow as they decide which pretrained model they would use in their project. So can you think about the last time when you choose a model from a model hub? So how did you choose it? Can you summarize your decision making process?

Speaker 2:

Before fine tuning it you mean?

Speaker 1:

It can include like the fine tuning process as well?

Speaker 2:

Okay. Usually what I go for is in order to find a solution to any of the problems we're facing at our startup, I usually look into different sources that could be sent like archive or a Google scholar that could be medium articles. That could be the model hub, where I try to find either data or a model for something that seems useful for us. And then when it's data I need to check with which model would be good to use it with. So for text generation, you could use any gpt model usually, or yeah, if you're like as I said earlier, transform keywords into sentences, then text to text models, like T5 are useful. I learned the different capabilities of different pre-trained models, such as T5, GPT, BERT, et cetera in my research in different boot camps that I did. And through that, I select them based on experience or start with my experience and then fine tune it or try alternatives if I see better fit or try to improve them even more.5

Speaker 1:

Where do you usually select the pretrained models? Do you select the models from the model hubs or you select the models from the GitHub open source projects?

Speaker 2:

Like if it's a paper, then they usually give the source of the model. Then I would use this one and sometimes ... so there's an [inaudible 00:04:25] model from Stanford that I used two years ago and I would host it on their own servers, but then oftentimes it comes from a model hub such as HuggingFace, where you can use the filters to pre-identify your tasks or even language slash framework, such as [inaudible 00:04:42] Flow, Flex, et cetera.6

Speaker 1:

Okay. So why you select model? Do you care more about the model's performance than the architecture?

Speaker 2:

Usually, initially, I care about the architecture and that it helps me in fulfilling my functionality. And then I know that there's different ways in order to improve the performance.7

Speaker 1:

Okay. Do you think the pretrained models available in model registries can accurately describe their behaviors?

Speaker 2:

I mean, if you look at T5 and I think there's subsequent releases that they did, where they trained them on 27 different downstream tasks or so, like a lot, at least, usually it's not in this rich detail in the model hub. So oftentimes you find it in your original paper when they're uploaded by practitioners, such as me or there's one guy from Spain that has uploaded like 120 something models, then you see that oftentimes there's not extensive documentation around the model, how it was pretrained what test scores were achieved, et cetera.8

Speaker 1:

Okay. To what extent do you think the discrepancies of the performance metrics can affect your decision making? So here for the discrepancies, I mean sometimes the actual performances can be different from the [inaudible 00:06:16] performance in the model cars or the documentation.

Speaker 2:

Yeah. I mean, for this, we usually run tests. We have a third person on our team that is not technical and usually we expose them to ... like once we have a model and we have it fine tuned, or we know which task we want to solve then we have like our pre defined test cases. And then I use some additional test cases and then he does extensive testing. I didn't mention him earlier because he doesn't really write code or anything, but like he tests it and then judges it from his communications background that he has.9

Speaker 1:

Can you elaborate more about how you test it?

Speaker 2:

Yeah, so first of all, we have like our own test set, you know where we say, "Okay, let's say we want to combine two sentences into one as with P5 and the past data ... " Another example there's paraphrasing, which you can use the past data set to find tune T5 for instance, and then it learns to paraphrase. And then we would take sentences and see if they are paraphrased once we run it through and then we would see if we're happy with that. And if we are, then we would usually take this as our validation data and then we would fine tune it, improve it, try different approaches to see if we score better or not. And actually, we created a system and in backend that allows us to score them, to classify whether we're happy, like it's better or worse. It's like five different categories you can choose. This is usually done by me and by our communications person.

Speaker 1:

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

Speaker 2:

We don't really attract robustness, so I can't really make comprehensive answer to that.10

Speaker 1:

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

Speaker 2:

I mean, if you have a very narrow use case, then explainability is usually pretty high because you understand okay, I put two sentences in there. One comes out it's merged and then sometimes it's harder to explain which conjunction was used. If a temporal connector or like a causal connector was being used, because that's what the model figure is out itself. Sometimes it's wrong, but oftentimes it's really good, so 95% acceptance rate. So yeah, we usually do this with plausibility and against our test set or our own validation set.11

Speaker 1:

Okay. So how frequently do you retrain the models?

Speaker 2:

We know that we have good models with good data. Everything coming from like big corporations, such as Google or Facebook where you know that the data sets are carefully cleaned and really easy to take and put into the fine tuning or transfer learning algorithms. And then in this case you do it actually kind of once or twice and then you're pretty happy with it. And then there is like data sets that we create ourself. Let's say this morning I realized the tools I looked at like keywords to create a text didn't or the models that were out there are not very persuasive from my perspective.

I'm considering to ... and I found, I think another scientific article where someone really professionally extracted the major topics of sentences. I'm considering right now, if I take a lot of sentences that I have, extract the major keywords and then reverse the data set. So I have the sentence, I extract the keywords and then I would show to T5, the text to text architecture first, the keywords and then the sentence. And then it would usually figure out itself how to operate on that. So yeah, this is how we do it. And in this case we do more of [inaudible 00:11:07].

Speaker 1:

Okay. So next question is how frequently do you finetune models?

Speaker 2:

This could be on a weekly basis, and then when we have a lot of traffic done. Yeah, it always ... so every 10,000 inputs that we get from users, we automatically leave and retrain them.12

Speaker 1:

Okay. Do you think the lack of trainability or fine tunability is a problem when reusing a pretrained model from the model registries?

Speaker 2:

Well, HuggingFace does provide with their [inaudible 00:11:45] auto train. They actually provide a retraining more or less set up, but then there's other infrastructures that help you. We have our script, so it's not really a problem.13

Speaker 1:

Okay. Is there any other challenges you met before when selecting a pretrained model from a model registries?

Speaker 2:

Yeah, I guess oftentimes it's over promising. So you read the documentation and you expect something, but then like the results are not really there. That's why it's so important to test at least with 10 examples and see how it goes and then to understand. Okay. Can I even use that? And then if you figure out that this is not really usable, then you would either come up with a different solution to the approach or you would try to create yourself what the person was writing about.14

Speaker 1:

Okay. Then we'll move onto the next set of questions, which is about the different software attributes. We want to learn more about what sort of information is useful to engineers who use the pretrained models. So here I have some traditional attributes. These attributes are defined by the MPM, which is used for the JavaScript packages. Can you take a brief look at the first several sentence or paragraphs of each attribute here? And let me know when you're already?

Speaker 2:

I'm done with quality. Do you want me to continue with popularity and maintenance?

Speaker 1:

Yeah.

What do you think would help your team's pretrained model from the model registries in terms of these three attributes?

Speaker 2:

Should I go in the sub qualities or should I be by the quality ... Oh, sorry. Should I go into the bullet points? Or do you want me to just go for the three?

Speaker 1:

Just the three, the top ones?

Speaker 2:

I mean, to be fair, all of them are helpful, I guess, quality and popularity a little bit more helpful in terms of ... I believe at least of models because there's not really open issues, total issues usually with models. I mean there are some merge requests on the end of changing documentation or something or making it more comprehensible, and it's also hard to collaborate on fine tune model, I guess. You just don't change weights.15

Speaker 1:

Okay. So here we define the following [inaudible 00:15:09] specific attributes and I will ask several questions about each of them. So first we'll talk about the problems here. We defined as a measure of model lineage or traceability. So some examples like link to the paper, whether it contains GitHub page or data sets. So can you think about time when you made the Providence problems when using the pretrained model before?

Speaker 2:

Could you please repeat your question?

Speaker 1:

So can you think about the time when you made the provenance problems before? Can you tell me the challenges you had?

Speaker 2:

Yeah, well oftentimes there is no paper, so usually I mostly look on Google data sets or on HuggingFace, such as that you know what we're talking about. So often there is no paper affiliation. So usually, their GitHub pages are not necessarily there. Data sets, architecture. This is usually known because they tell you which architecture they use oftentimes. Yeah.16

Speaker 1:

So what do you think would be useful to know beforehand in order to solve the provenance problems?

Speaker 2:

I mean, you can't force people to write papers necessarily. I do not have like a solution for this right now. I mean examples would always be really great, but I'm not sure if they would be in the other two categories. Doesn't look like it though.17

Speaker 1:

Okay. Then we'll move onto the reproducibility. So here we define it as the ability of a [inaudible 00:17:05] practitioner to produce the same accuracy and training or evaluation time from pretrained model as defined in its paper, source code or the groups. Can you think about time when you met any reproducibility problems when using the models before?

Speaker 2:

Yeah, the second, like for instance ... so some neural nets only work with a given prompt, so you need to ... particularly when they fine tuned multiple downstream tasks, and then if this prompt is missing, then you do not know how to use it. This can be quite a big problem then. So again, as an example, if you want to paraphrase, then you would, for instance, have a prompt paraphrase call in space and then you would insert your sentence that you want to paraphrase. And if this isn't there, then neural net won't work. So this could be one of them.18

Speaker 1:

Okay. So what do you think would be useful to know beforehand in order to solve the reproducibility problems?

Speaker 2:

Clear documentation on this regard and yeah, you wrote it down there, a demo as a notebook usually is good. Yeah. Because it just helps to understand how you should use the code.19

Speaker 1:

Okay. The last attribute here is portability. So here we define as is with which an engineer can take a pretrained model and reuse it in another environment or another software project. Can you think about a time when you met any possibility problems when using the trend models before?

Speaker 2:

Yeah. This would usually relate to ... we had a few model setups where we would have some interesting dependencies or difficulties with some frameworks in order to make them run. I don't know, for instance, if you know, spaCy, which is allows to also do by now deep learning in [inaudible 00:19:09] NLP context, but yeah, setting environmental parameters or deploying more sophisticated concepts such as spaCy can face difficult ... like one can face difficulties in doing that and then you need to define them when you deploy them for instance to a cloud framework.20

Speaker 1:

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

Speaker 2:

Well, in this regard, it would really help if some would like provide perfectly written code as a docker container, so you can just deploy it. But obviously this is a so wishful thinking, so yeah, I guess you need to get training in cloud deployment to deal with it and Google.21

Speaker 1:

Okay. So except for this three attributes, do you think there are any other attributes would be helpful for the pre trend model you're using?

Speaker 2:

Yeah. I mean you wrote notebook demo and inference demo. What I really started to like recently is I don't know if you're familiar with Gradio or streamlet which that give you kind of a live demo approach and allow you for flagging, et cetera, where you can like radio also flagging. At least if you deploy it on your local machine and then through this, you can kind of get feedback from colleagues, et cetera, much easier and quicker. And so something like this. Like other people use this examples and liked this or that, or kind of it's kind of like Amazon's review on any item you could buy, one star to five star. Something like this, I guess, would help to improve the hub if that makes sense.22

Speaker 1:

Okay. So the last set of questions is about the trustworthiness of the pretrained models. So we are trying to understand how the pretrained model shortcomings affect engineers ability to rely on and reuse them. The first question here is which aspect of the pretrained model do you assume are trustworthy?

Speaker 2:

Well, only the results that I see when I run my examples. So I always have to test it myself manually in order to have trust on what is happening there. Yeah. And then obviously if you have ... like here, we see the deep learning specific attributes, if the provenance column is filled out, so you have a link to the paper, you have the affiliation of let's say a big this American corporation, then you obviously have a little bit more trust than from individuals from certain places in the world that ... this is I guess where you can like have different trustworthiness from [inaudible 00:22:32].23

Speaker 1:

OK. Yeah. Have you ever found any discrepancies between the 10 pretrained models and the downloaded version? So in terms of the accuracy latency architecture or some other things.

Speaker 2:

Yeah, I guess sometimes authors do not provide the data they use the trained model with, so they only provide the model and then you don't really know how they achieved the great success rates because it doesn't score as persuasively as you expected it to do or like you know ...24

Speaker 1:

How did you find the problem and how did you address these problems?

Speaker 2:

This actually was a research paper in January 2021. First of all, the code was, I had a lot of errors. So I had to make a lot of changes in the code. And then the results that they presented in the paper were not matching the quality of the results that I got, even when I was using the examples. And so in order to fix that, I tried to make more improvements and didn't realizing that there is no ... it doesn't achieve what I wanted it to do. I went to another solution.25

Speaker 1:

Okay. To what extent do you think these performances are acceptable?

Speaker 2:

What I learned from our third guy that doesn't understand code, he wants 100% accuracy results, because he thinks that AI can solve any problem, but he's usually happy when you are in the 90% plus score like that he doesn't reject, that he only rejects, let's say one or two examples out of 20 that we give him. That's kind of the acceptability that we need to achieve.26

Speaker 1:

Okay. The last question here is, do you think the discrepancy will have significant impacts like on your product?

Speaker 2:

Well, there's so much out there. It's really the difficulties to find good quality code or data. Once you do that, you're actually in the green. It just kills time, I guess, to go through examples that don't work.27

Speaker 1:

Okay. Okay. That's all of my questions. I will stop recording though. Thank you very much for your time.

**Annotations**

1 Role:

- ML engineer works together with a SW engineer

- Deploying a tool on GCP

- Try and fine-tune our transfer learning DNN to teach them downsteam tasks in order to apply them to text.

- Then I would like to deploy them on GCP as microservices by a fast API in Python.

- Then our SW engineer integrates them into our front end.

2 Reuse scenario:

- Interested in figuring out how I can use keywords in order to generate a text from it.

- Trying different approaches and I found a solution that someone used and he uploaded the model to HF.

- I downloaded it from HF and ran some inference and experiments to see whether that's useful for us or not.

3 Fine-tune?

- I create artificial datasets or I use datasets that are out there to fine tune models.

4 Backbone?

- Yes

5 Decision making:

- look for articles from different sourcse that could be sent like arXiv or Google scholar, or model hub

- Then find the task and the relevant models

- Learn the different capabilities of different PTMs

- Then select the model based on experience or start with my experience and then fine tune it or try alternatives if I see better fit or try to improve them more.

6 Where?

- Paper -> original source

- Pre-identify your tasks or even language framework -> model hubs

7 Performance or architecture?

- Architecture!

- Iit helps me in fulfulling my functionality

- I know different ways to improve the performance.

8 Discrepancy?

- Example: T5

- Subsequent releases

- They trained on 27 different downsteam tasks or so. Usually it's not in this rich detail in the model hub

- Often there is not extensive documentation around the model, how it was pretrained, what test scores were achieved, etc.

9 Discrepancy:

- We usually run tests.

- We have our pre-defined test cases.

- Then we use some addtional test cases and extensive testing.

- Another guy does not write code or anything. He tests it and judges it from his communication background.

- We created a system and in backend that allows us to score them.

10 Robustness:

- We don't really attract robustness.

11 Explainability:

-Very narrow use case -> explainability is usually pretty high because you understand okay

-

12 Retrain:

- We know we have good models with good data.

- Everything comes from Google or Facebook.

- Really easy to take and put into the fine tuning or transfer learning algorithms.

Fine-tune:

- Weekly basis.

- We have a lot of traffic done.

- Every 10000 inputs that we get from users, we automatically leave and retrain them.

13 Trainability/Fine-tunability:

- Huggingface provides a retraining setup, Autotrian

- There is other infrastructures that help you. We have our script, so it's not really a problem.

14 Other challenges:

- Over promising. The results are not really in the documentation.

- It's so important to test at least with 10 examplse and see how it goes and then to understand.

- If it's not really usable, you would either come up with a different solution or create yourself what the person was writing about.

15 Traditional attributes:

- All of them are helpful

- Quality and Popularity:

- There are some merge requests on the end of changing documentation or make it more comprehensible.

- Maintenance: you just don't change weights.

16 Provenance issue:

- Oftentimes there is no paper affiliation.

- Usually their GitHub pages are not necessarily there.

- Datasets, architecture are usutally known because they tell you.

17 Provenance help:

- You cannot force people to write papers

- I don't have a solution for this right now.

- Examples would always be really great.

18 Reproducibility issue:

- Some NN only work with a given prompt.

- If the prompt is missing, then you do not know how to use it. This can be quite a big problem.

- If you want to paraphrase, you would have a prompt paraphrase call in space and then you would insert your sentence. But if this isn't there, then the NN won't work.

19 Reproducibility help:

- Clear documentation

- A demo as a notebook (just help to understand how you should use the code)

20 Portability issue:

- Setting environmental parameters or deploying mroe sophisticated concepts can face difficult.

- You need to define them when you deploy them for instance to a cloud framework.

21 Portability help:

- If some would like to perfectly written code as a docker container, so you can just deploy it.

- Obviously this is a so wishful thinking. You need to get training in cloud deployment to deal with it and Google.

22 Other attribute:

- Demos: inference, notebook, live demo

- You can get feedback from colleagues, etc. much easier and quicker.

23 Trustworthiness:

- Always have to test it myself manually in order to have trust on what's happening there.

- Provenance -> a little bit more trust than individuals

24 Discrepancies:

- Some authors do not provide the training data. They only provide the model and then you don't really know how they achieved the greate success rates

25 How did you find discrepancies:

- The code, I had a lot of errors. So I had to make a lot of changes in the code.

- The results they presented in the paper were not matching the quality of the results I got, even when I was using the examples.

- I tried to make more improvements. But eventually I went to another solution.

26

27 Significant impacts?

- So much out there.

- Reallt difficult to find good quality code or data.

- Going through examples that don't work just kills time.