Interview Protocol

# *Consent form: <https://purdue.ca1.qualtrics.com/jfe/form/SV_6Xw3lmOuVAG9y2a>*

*(Some demographic questions are included in the consent form.)*

# Research Questions

* + RQ1: What are the challenges and practices for PTNN selection?
  + RQ2: What attributes can be added to improve the model registries?
  + RQ3: Do the PTNNs available in model zoos accurately describe their behavior?

# Interview Questions

This interview is related to the trustworthy re-use of ***Pre-trained Neural Networks (*PTNNs*)***. Questions fall into 5 categories:

Background, Model Selection, Attributes, and Trustworthiness.

# 

# Demographic Questions (~5 min)

First, I would like to ask some questions regarding your background. The background questions can be answered briefly so that we can save time for the other interesting questions.

1. Could you briefly describe your role in your team?  
   Answer: TechLead1
2. Can you tell me a recent time when you used a pre-trained model from an external model hub?  
   Answer: I needed to create a machine learning model on natural language processing for my job and downloaded and used a pre-trained model on huggingface as a base.
   1. What challenge did you meet?  
      Answer: I didn't have any particular challenges.2
   2. How did you address it?

# Model Selection (~10 min)

The next set of questions is related to how you select a pre-trained neural network (PTNN). We are trying to understand the process that software engineers follow as they decide which PTNN to reuse in their projects.

1. Think about last time when you chose a PTNN from a model hub. How did you choose it? Can you summarize your decision-making process?  
   Answer: Since the model was to be used as a base for fine-tuning, we did not choose from so many options, but we chose one that met the requirements of the service. We chose it based on cost, latency, and no licensing issues.3
   1. Where do you select the PTNNs,
      1. pretrained models from a model hub or models serialized in github?   
         Answer: from a model hub
      2. Which hub do you choose to use?   
         Answer: huggingface
      3. Why?  
         Answer: huggingface provides the library named “transformers” so the hub is easy to use.4
   2. When selecting the model, do you care more about the model’s performance than the architecture?  
      Answer: Yes. I consider it essential to model's performance, such as accuracy or how the cost is needed to run, because I’ve to keep costs down and high accuracy to provide service to customers.5
2. Do you think the PTNNs available in model registries accurately describe their behavior?   
   Answer: No. I think there are many uncertain models. Because some models cannot reproduce their performance6.
3. To what extent do the **discrepancies of performance metrics** *(i.e. the actual performances are different from the claimed performances)* affect your decision-making?   
   Answer: That is a minor issue. I don't count on it from the beginning that the base model's claimed performance which got from the hub because I finetune it eventually.7
4. To what extent does the **robustness** of the models affect your decision?  
   Answer: That is a little important. Because I need robustness to the final production model which I service, I don't need it for pre-trained models. I don't expect robustness in the pre-trained model.8
5. To what extent does the **explainability** of the models affect your decision?  
   Answer: That is important for my decision because the machine learning model is required an explanation, such as what data are included, what kind of personal data is contained, or how the process is executed.9
6. How frequently do you retrain the models (e.g. for PTNN validation or model adaptation/transfer learning)?  
   Answer: About once a month.
   1. Do you think lack of trainability is a problem when reusing a PTNN?  
      Answer: If I can't train at all, that's a problem, but if the means of training are not mentioned, I don't particularly care because that is something I will figure out on my own.10
7. When selecting a PTNN from model registries, are there any other factors you consider?  
   Answer: Consider ease of use. In other words, we are concerned about whether it is possible to continuously obtain and finetune the pre-trained model and whether the model will suddenly disappear. In this respect, huggingface is characterized by a greater sense of security than models uploaded by individuals.11
8. What other challenges do you face when selecting a PTNN implementation from a model registry?  
   Answer: It isn't easy to use a model that does not describe what data was used or how to use it in the first place. In particular, the license is an essential element; if it is not written, it is very bothered to use. Even if the license is written, it is very bothered to be distributed under a dangerous license such as Creative Commons SA or GPL.12

# Attributes (~10 min)

This set of questions relates to DL software attributes. We want to learn about what sort of information is useful to engineers who reuse PTNNs.

***(Show the slides)***

|  |
| --- |
| NPM defines the following attributes for Javascript Packages: |

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. What do you think would best help your team select a pre-trained model from model registries?  
   Answer: Popularity. Because I already referenced popularity such as huggingface's download counts or stars. Popularity means so many people use it. This means that there is likely to be a lot of information about that model on the Internet, making it readily available.13

We define the following attributes:

## **Provenance**

*A measure of model lineage or traceability.*

1. Tell me about a time when you met **Provenance** problems when using PTNNs before (e.g. link to the paper, research prototype, GitHub page, custom website)?  
   Answer: The license issue I faced is maybe the Provenance problems you said. I'd like to know what data is contained, how created, how licensed, and under law.
   1. \*How did you address the problems?  
      Answer: For models for which the license was not clear, I recreated them myself using the same data.14
2. Do you have other specific **Provenance** problems?  
   Answer: No, I haven't any other problem with the Provenance.15
3. What would have been useful to know beforehand in order to solve those problems?  
   Answer: Knowledge of the law. In particular, we need to learn about the country's laws where the servers actually run the machine learning, as privacy and copyright issues in machine learning data vary from country to country16.

## **Reproducibility**

*The ability of a DL practitioner to produce the same accuracy and training/evaluation time from a PTNN as defined in its paper, source code, or group.*

1. Tell me about a time when you met any **Reproducibility** problems when using PTNNs before (e.g. runnability, configuration, data type, accuracy variance)?  
   Answer: When I reproduced the model, I didn't understand how accurate I should aim. I'd like to know beforehand what accuracy point is good because the pre-training needs a long time. I want to model creator to write the final score on README.
   1. \*How did you address the problems?  
      Answer: I continued the pre-study to the point where the inference results from the model were close.17
2. Do you have specific **Reproducibility** problems?  
   Answer: Machine learning also depends on machine specs, so please also write down machine specs to ensure reproducibility. Also, learning with TPUs, for example, can never be reproduced on consumer GPUs, so I feel that such areas are a challenge for reproducibility in machine learning.18
3. What would have been useful to know beforehand in order to solve those problems?  
   Answer: Although it is a detail, it is important to understand the impact of batch size on the final model, etc., in order to reproduce the model. It is also advisable to confirm the level of accuracy that should be aimed for in the pre-training for that model architecture before reproducing the model.19

## **Deployment Constraints**

*The ease with which an engineer can take a PTNN and reuse it in another environment, software project, etc.*

1. Tell me about a time when you met any **Deployment Constraint** problems when using PTNNs before?  
   Answer: BERT has extraordinary high accuracy, but BERT is very very heavy. Using BERT on production env is too difficult because production env hasn't many machine resources. I gave up using BERT.20
2. Do you have specific deployment constraints? For example, latency, fine-tunability, energy consumption, accelerator  
   Answer: I often have constraints such as response latency when inference, deployable model size, cannot using GPU or a few machine resources. Because creating service has many constraints on costs and customers' needs.21
   1. \*How did you address the problems?  
      Answer: The first step is to review the model selection. Does it really have to be that model architecture, can it be solved without using DeepLearning? etc. If the model cannot be changed by any means, consider using a special instance, such as Inferentia from AWS.22
3. What would have been useful to know beforehand in order to solve those problems?  
   Answer: It is a rare case to use GPU for inference on production env. We need to know what the cost is to use that model, how much latency is incurred for GPUs, what about CPUs, and so on.23

**Others**

Except for these three attributes, do you think there are any other attributes that would be helpful for PTNNs? (e.g. robustness, explianability)  
Answer: Just having the three indicators you defined would be very helpful. I hope this can be achieved.24

# Trustworthiness (~5 min)

This set of questions is about PTNN trustworthiness. We are trying to understand how PTNN shortcomings affect engineers’ ability to rely on and reuse them.

1. Which aspects of the PTNN do you assume are trustworthy (e.g. the model architecture matches the paper; the performance with the weights is correct, ...)?  
   Answer: model's architecture and accuracy25
2. Have you found any discrepancies between the claimed pre-trained models and the downloaded version (e.g., accuracy, latency, architecture)?  
   Answer: I have never faced such a contradiction, especially until now. This is because the models I use are mainly created by research institutes and universities. When I use private models, I am aware that the accuracy may not be correct.26
   1. How did you find them?
   2. How did you address this problem?
   3. To what extent do you think these discrepancies are acceptable? (e.g. less than a certain percentage discrepancy of accuracy)  
      Answer: I don't think it is pardonable to have different architectures. As for accuracy, I think a margin of error of 10% should be tolerated.
   4. What would you do if you find a model with discrepancies?  
      Answer: I want to report it to the author, but it is difficult without an easy place to report it.27
3. Do you think the discrepancies will have significant impacts? (For example, if the model is used in safety-critical systems.)  
   Answer: It depends on what kind of discrepancies, but the impact is sometimes serious. For example, if the model architecture is different from what it claims to be, and there is one layer in the mix that is not supposed to be there, can you trust what that layer is doing? I would be afraid to use that model in a production env. Because the DeepLearning model is a black box, I want the claimed architecture to match the actual architecture.28

# 

# *Anonymization*

*Audio from the interviews will be recorded and transcribed by a trusted 3rd party service. The transcriptions will be viewed, and personal information will be anonymized. The audio recordings will be deleted after the transcripts are received, reviewed, and anonymized. A key mapping participant to the transcript will be kept separately by the research team until the conclusion of the study – this key may inform further participant selection, e.g. further probing of an interesting organization.*

**Annotations**

1 Subject is a Tech Lead

2 Subjected used a PTNN as a base for a ML model doing NLP

Subject encountered no “particular challanges"

3 Last PTNN was chosen based on:

Cost

Latency

Lack of licensing issues

4 Model was chosen from a model hub (huggingface)

Huggingface was chosen due to a library that made finding a model easy

5 Architecture is more imporant than performance becuase it is essential to performance.

6 Any models can not reproduce their performance

7 discrepancies of performance metrics is a minor issues since is will be fine-tuned

8 Robusness is “a little important” but not expected in PTNN

9 Explainability is important to subject decision.

10 Models are trained once a month

If a model can't be trained that is an issue but if the documentation is lacking the subject can figure it out

11 Ease of use is concidered when selecting PTNNs

Support and finetunability are important

models uploaded by individuals are suspect

12 Licensing and lack of data desrciption are challenges faced when implenting a PTNN

13 Popularity is the most useful metric for PTNNs

14 Licensing is a providence issue.

If the license is not clear the model has to be recreated

15 Subject has had no other providence issues.

16 Understanding of related laws would have been useful to solve providence issues.

17 Reproducibility became an issue when the models accuracy was not made explicit.

Subject had to guess when the results were "close"

18 Different hardware (TPU vs GPU) and lack of specs create reproducibiltiy issues

19 Impact of batch size and level of accuracy should be explicity stated to reduce reproducibility issues

20 Subject gave up on BERT due to a deployment constraint - not enough resources

21 Latency, model size and not being able to access GPU are all deployment contraints

22 Deployment constraing issues are solved partially by model selection and if needed changing the deployment hardware.

23 Latency for GPUs and CPUs would be good to know beforehand to avoid deployment constraint issues

24 The three indicators mentioned would be very helpful.

No other indicator is metnioned.

25 A model's accuracy and architecture are assumed to be accurate

26 Subject has not observed a discrpency between claims and downloaded models

Subject avoids private models in favor of models uploaded by institutins since private models accuracy may not be trustworthy

27 10% margin of error shoud be tolerated in accuracy discrepencies

Ideally discrepancies should be reportable

No easy place to report discrepencies exists

28 Discrepancies can be serious.

Example is given of a incorrect architecture being described.