Technical Report - Applying Revised Machine Learning Issues Taxonomy

Tuan Dung Lai, Anj Simmons, Scott Barnett, Jean-Guy Schneider, Rajesh Vasa

I. Introduction

In this technical report, we describe the iterative approach used to label our machine learning (ML) issues dataset against the taxonomy proposed by Humbatova et al. Humbatova et al. (2020). Our aim is to revise the taxonomy by Humbatova et al. and develop a robust protocol for consistently applying the taxonomy, ensuring the reliability and replicability of our research results. This process is an integral part of our research Lai et al. (2022), which focuses on comparing the differences between ML issues and non-ML issues. We will randomly sample 30 issues from our dataset of 147 ML issues, and the first three authors will independently label the issues using the taxonomy. Subsequently, we will calculate the inter-rater reliability to assess the agreement among the three raters. We will repeat the process and refine the taxonomy until we achieve a moderate level of agreement.

II. Background

We validate our ML issues dataset against an existing taxonomy of real faults in deep learning (DL) systems by Humbatova et al. (2020). The paper introduces a large taxonomy of faults in DL systems containing five top-level categories, three of which are further divided into inner subcategories. The authors manually analysed 1,059 artefacts gathered from GitHub commits and issues of projects that use the most popular DL frameworks (TensorFlow, Keras, and PyTorch) and from related Stack Overflow posts. They also conducted structured interviews with 20 researchers and practitioners describing the problems they have encountered in their experience, which enriched their taxonomy with a variety of additional faults that did not emerge from the other two sources. The final taxonomy was validated with a survey involving an additional set of 21 developers, confirming that almost all fault categories (13/15) were experienced by at least 50% of the survey participants. The paper uses faceted classification, i.e., it created the categories/subcategories of its taxonomy in a bottom-up way, by analysing various sources of information about DL faults.

III. Method

We manually label ML issues in the ML issue dataset against an existing taxonomy Humbatova et al. (2020) using an iterative approach. We randomly sampled 30 ML issues from the ML issues dataset. Three researchers independently applied the taxonomy on the 30 issues and calculated the inter-rater reliability. We repeated the labelling process and made adjustments to the taxonomy until a moderate level of agreement was reached (Kappa score above 0.61 Landis and Koch (1977)). The inter-rater agreement of the labelling process is measured by the Light's Kappa metric Light (1971), which equals the average of all possible combinations of bivariate Kappas between raters. This metric indicates an overall index of agreement. A low Light's Kappa metric will be addressed, by refining the taxonomy and repeating the process until the agreement is substantial. After that, the first author independently labelled the rest of the ML issues dataset.

In the first iteration, three researchers independently labelled a random sample of 30 ML issues using the issue title, description, comments, pull requests, file changes and repository descriptions. This resulted in no agreement using the inter-rater reliability test Landis and Koch (1977) (Kappa score = .15). After that, we discussed each issue and come to a consensus, we modified the definitions of the taxonomy to avoid ambiguity in the second iteration, merging all subcategories in the "Tensor and Inputs" categories because they are hard to distinguish. After that, we run the labelling process again on another set of 30 random sampled ML issues, in which 2 issues were presented in iteration 1. This resulted in a weak agreement (Kappa score = .43)

The first two labelling iterations resulted in low Kappa scores which is evidence that taxonomy is inadequate to be used for real bugs in applied AI projects. In Humbatova et. al's study, the authors show the creation process of the taxonomy but do not show how the taxonomy is validated or applied in practice. For those reasons, we narrowed down the number of categories to 6 and also add a protocol for the labelling process: First, open the issue URL, read the issue title, issue description, comments, code change, PR, and project description to label, then go through the list of categories from top to bottom (as indicated in the list below) and pick the first one that is a match. Enhancement and non-critical issues (issues that do not crash the program) can be labelled using one of the 6 categories. Code refactoring can be in the first 5 categories, if not suitable, label as "Other". The third iteration of the labelling process was conducted on another set of 30 randomly sampled ML issues (all of the 30 issues are excluded from the first 2 iterations), this iteration resulted in a moderate agreement among the three raters (Kappa = .67). The new definition of the 6 categories is described in Table II.

	Iteration 1	Iteration 2	Iteration 3
Number of categories	19	15	6
Number of issues	30	30	30
Kappa score	.15	.43 (Weak)	.67 (Moderate)

TABLE I: Inter-rater reliability between 3 raters when labelling ML issues

Categories	Definitions
GPU Usage	Incorrect or inefficient usage of GPUs, wrong reference to GPU device, failed parallelism, incorrect state sharing between subprocesses, faulty transfer of data to a GPU device.
Model	Inappropriate, inefficient or incorrect model initialisation, choice of architecture. Inappropriate use of activation function, incorrect properties for a neural network layer, missing, redundant or wrong layer. Errors occur during inference.
Tensor and Input	Error or inefficiencies in data quality such as low-quality data, noisy data, imbalanced data, and insufficient data. Inappropriate preprocessing of data such as scaling, normalisation, and feature engineering. Incorrect shapes of input, wrong dimensions, size, inappropriate file type, encoding, and selection of data format.
Training Process	Inappropriate or inefficient training processes excluding data-related problems, such as inappropriate batch sizes, and learning rates. Hyperparameter issues such as learning rate, dropout rate, and number of epochs. Inappropriate optimiser, inappropriate choice of loss function when using during training. Inefficient or incorrect validation/ testing procedure.
Third-party Usage	Inappropriate usage of third-party programs or libraries, such as TensorFlow, PyTorch, Keras, and Numpy.
Other	Documentation issues or anything unrelated to the 5 categories above.

TABLE II: Extended definitions of 6 ML issue categories

Compared to the original taxonomy, we have changed “API” to “Third-party usage” because some issues belong to other libraries, we generalise it to cover more cases. Additionally, data quality issues such as wrong data format and preprocessing occurring in the training process can either be labelled in the “Training” or “Tensor and Inputs” category, for this reason, we pull data-related categories “Training data quality” and “Preprocessing of training data” out of their original parent category “Training” and add them to “Tensor and Input”. For clarity, we change “Training” to “Training Process” to emphasise that the issues must happen in the process of training.

IV. Results

Table I shows the Kappa score from each labelling iteration. After two iterations, we were able to achieve a moderate level of agreement among the three raters (Kappa = .67).

V. Taxonomy

Table II shows the extended definitions of the revised taxonomy.

VI. Conclusion

In conclusion, we were able to establish a robust protocol to apply the revised taxonomy. The revised taxonomy with the new definitions can be used to reliably label ML issues from open-source applied AI projects on Github. The revised taxonomy has 6 categories, documentation, enhancements and non-critical issues can be classified using the taxonomy. Our labelling protocol indicates that the first category from the list matches the issues, the higher level category will be more favoured than the one below it. Based on the purpose of the research, a Multi-label protocol can be used to eliminate this issue.

References

Nargiz Humbatova, Gunel Jahangirova, Gabriele Bavota, Vincenzo Riccio, Andrea Stocco, and Paolo Tonella. 2020. Taxonomy of real faults in deep learning systems. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. 1110–1121.

Tuan Dung Lai, Anj Simmons, Scott Barnett, Jean-Guy Schneider, and Rajesh Vasa. 2022. Comparative analysis of real bugs in open-source Machine Learning projects–A Registered Report. *arXiv preprint arXiv:2209.09932* (2022).

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

Richard J Light. 1971. Measures of response agreement for qualitative data: some generalizations and alternatives. *Psychological bulletin* 76, 5 (1971), 365.