Ensuring FAIRness in Machine Learning Projects

Şefika Efeoğlu[1,2,†], Zongxiong Chen[3] and Sonja Schimmler[1,3]

[1] Technische Universität Berlin, Germany
[2] Freie Universität Berlin, Germany
[3] Fraunhofer FOKUS, Berlin, Germany

Abstract

Subsymbolic approaches like machine learning (ML), deep learning, and Large Language Models (LLMs) have significantly advanced Artificial Intelligence, excelling in tasks such as question answering and ontology matching. Despite their success, the lack of openness in LLMs' training datasets and source codes poses challenges. For instance, some ML-based models do not share training data, limiting transparency. Current standards like schema.org provide a framework for dataset and software metadata but lack ML-specific guidelines. This position paper addresses this gap by proposing a comprehensive schema for ML model metadata aligned with the FAIR (Findability, Accessibility, Interoperability, Reusability) principles. We aim to provide insights into the necessity of an essential metadata format for ML models, demonstrate its integration into ML repository platforms, and show how this schema, combined with dataset metadata, can evaluate an ML model's adherence to the FAIR principles, fostering FAIRness in ML development.

Keywords

Machine Learning, FAIR ML, ML metadata

1. Introduction

Subsymbolic approaches such as machine learning (ML), deep learning, and recently, Large Language Models (LLMs) have illustrated outstanding advances in Artificial Intelligence. LLMs have achieved remarkable results in downstream tasks, such as Question Answering [1], Ontology Matching [2] and Image Caption Generation [3]. Recent research in these downstream tasks uses either general-purpose LLMs directly or fine-tunes them with specific datasets.

FAIRness plays an important role in the repetition of experiments in scientific research. Wilkinson et al. [4] propose a guideline that clearly explains the Findability, Accessibility, Interoperability, and Reusability (FAIR) principles. To satisfy these principles, we should ensure that approaches using LLMs provide their weights, source codes along with their settings, and datasets. In recent research, an ML-ready metadata format for datasets based on the dataset schema as part of schema.org called Croissant [1] [5] is proposed. The integration of it as a plugin in the Hugging Face platform provides metadata about the datasets used in ML models, such as

4th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment 12 November 2024 - Baltimore, MD, USA co-located with The 23rd International Semantic Web Conference, ISWC 2024

*Corresponding author.

sefika.efeoglu@{fu-berlin.de,tu-berlin.de} (Ş. Efeoğlu); zongxiong.chen@fokus.fraunhofer.de (Z. Chen); sonja.schimmler@fokus.fraunhofer.de (S. Schimmler)

0000-0002-9232-4840 (Ş. Efeoğlu); 0000-0003-2452-0572 (Z. Chen); 0000-0002-8786-7250 (S. Schimmler)

[1] Croissant web page https://research.google/blog/croissant-a-metadata-format-for-ml-ready-datasets/

their validation, train and test splits, size, and description. Additionally, to reproduce machine learning research, Vanschoren et al. [6] recommend that three requirements should be fulfilled all together: (i) open source software, (ii) open data, and (iii) open access paper. Therefore, ML-based models cannot be evaluated solely by checking if their software or datasets are open separately. To determine whether they are truly open or not, we should take into account all their components comprehensively.

1. **Datasets:** The Croissant metadata format has been developed for ML-ready datasets based on the property *Dataset* [2] under *CreativeWork* at schema.org in [5]. The Croissant metadata format provides all metadata necessary for using a dataset in an ML model and has been integrated into Hugging Face [3], a well-known platform in the ML field. **Challenges:** The dataset schema alone is not sufficient for evaluating ML models from a FAIRness perspective. Additionally, Raza et al. [7] introduce a pipeline for LLM FAIRness in terms of datasets. However, having solely open datasets is insufficient to achieve FAIRness in the development of ML models.

2. **Source Code:** With regard to source code, schema.org provides schemas under *CreativeWork* for *Software Application* [4] and *Source Code*[5]. **Challenges:** It cannot be directly used for ML-based software and should be extended with configuration and evaluation results, such as metrics for ML-based software. The performance of the ML models on a dataset is measured with metrics such as F1 score and precision, depending on the specific tasks. This should also be included in the ML schema to adhere to the FAIR principles.

3. **Models:** The model weights are learned during the training process. To facilitate the repetition of a model's evaluation on the test split of a dataset, the model must be open and accessible [8]. This transparency ensures that the evaluation process can be independently verified and replicated. **Challenges:** Since ML models consist of mathematical functions, e.g. normal or uniform distribution for model weight initialization and various sampling strategies for post-processing, the evaluation results might vary. To ensure clarity and reproducibility in experimental models, it is essential to explicitly state the mean, maximum, and minimum metrics used in the experiments, as well as the standard error and the number of repetitions for each experiment [6, 8]. Unfortunately, schema.org currently lacks metadata to represent these crucial details in its existing framework.

MLDCAT-AP has recently introduced the Machine Learning Model entity [6]; however, it lacks specifications for the hardware requirements crucial for ensuring the reusability of ML projects. There is no complete schema defined to represent all metadata (or terminology) of ML projects such as hardware requirements [8]. In this position paper, we aim to provide insights into the necessity of metadata for the ML projects. We further provide first guidelines on how

[2] Dataset of schema.org: https://schema.org/Dataset

[3] Hugging Face: https://huggingface.co/

[4] Software Application:https://schema.org/SoftwareApplication

[5] Software Source Code: https://schema.org/SoftwareSourceCode

[6] MLDCAT-AP: https://semiceu.github.io/MLDCAT-AP/releases/2.0.0/#MachineLearningModel

to evaluate the ML project according to the **F**indability, **A**ccessibility, **I**nteroperability, and **R**eusability (**FAIR**) principles [4] within a layered architecture in Section 2.2. In the remainder of this paper, we will first present existing metadata for ML models in Section 2.1, and then we discuss a layered architecture for an ML model metadata format in Section 2.2. Finally, we summarize the benefits and future directions for FAIRness in ML projects in Section 3.

2. FAIRness for Machine Learning

This section first evaluates and discusses existing metadata related to machine learning (ML) models, as presented in Section 2.1. Subsequently, it proposes a layered architecture for ML projects and explores its potential integration into an ML repository platform, as detailed in Section 2.2.

2.1. Existing Metadata for Machine Learning Models

This section first discusses which metadata can be included from *Software Source Code* and *Software Application* schemas of schema.org. Afterwards, we discuss what metadata available in Hugging Face model cards can be utilized.

Schema.org

Schema.org has already proposed vocabulary for *Software Application* and *Source Code* under *CreativeWork*. However, ML source codes include unique configurations that differentiate them from regular software source code. Therefore, these terms are not sufficient to represent the metadata of ML models, and an extension of their schema is required. We tried to classify which metadata from the *SoftwareSourceCode* and *SoftwareApplication* should be included in an ML schema, as can be seen in Table 1. However, there is no metadata in these schemas to represent the configurations of ML models, e.g., hyperparameters, evaluation metrics, and datasets. Therefore, these *SoftwareSourceCode* and *SoftwareApplication* schemas are not rich enough to represent the metadata of ML models.

Hugging Face Model Cards

Hugging Face provides three ways for including model-specific metadata in a model card [7]: (i) using the metadata user interface (UI) illustrated in Figure 1, (ii) editing the YAML section of the README.md file in a model card and (iii) via the huggingface_hub [8] Python library. We aim to combine the metadata available in the UI of Hugging Face (See Figure 1) with model configurations from the README in the model card. This README might include configurations such as hardware requirements, hyperparameters, and platform details for reproducibility. Solely taking into account the metadata UI is insufficient to determine if the model is reproducible. Additionally, we examined what metadata is semantically similar to the metadata in *SoftwareSourceCode* and *SoftwareApplication* schemas in Table 2.

[7] https://huggingface.co/docs/hub/model-cards#model-card-metadata

[8] https://huggingface.co/docs/huggingface_hub/index

2.2. A Layered Architecture for Machine Learning Projects

We propose a layered architecture for an ML model metadata format inspired by Croissant [5] to support both datasets and ML model source code across the following four layers:
Metadata Layer: This layer contains general information about the dataset and the ML model source code with its settings, including its name, description, runtime requirements, software requirements, and license. Both Table 2 and Figure 1 provide information about the existing metadata available in the schema.org format and on the Hugging Face platform.
Resources Layer: This layer describes dataset resources, the source code of the ML model, and the learnt model weights obtained during the training process using the dataset.
Structure Layer: This layer describes and organizes the structure of the resources, adopting the data structure defined in the Croissant ML-ready format [5]. Additionally, the configuration of the ML model, including the neural network architecture and corresponding hyperparameters (e.g., learning rate and weight decay), is detailed in this layer. The aim of the model configuration, including hyperparameters, is to support various ML frameworks, such as PyTorch [9] and TensorFlow [10], allowing for flexible network instantiation and evaluation. These ML frameworks are described as software requirements and are defined by library names on the Hugging Face Model Card (see Figure 1 and Table 2).
Semantic Layer: This layer bridges ML-specific data and model interpretations with semantics, describing the required metadata. The Semantic Layer details the requirements of ML models to repeat experiments, the datasets used in evaluations, tasks like text-to-text generation, and evaluation results, including metrics and their detailed values. It is designed to be extendable, catering to the evolving needs of the ML community and supporting domain-specific application endpoints.

Consequently, the aforementioned layers propose a comprehensive framework to evaluate the FAIRness of ML models within an ML project. Each layer defines a different perspective of the ML project with respect to its resources in terms of FAIRness. The ML project that provides metadata for these layers can be easily evaluated using various FAIR evaluation criteria, such as the evaluation system defined in [11].

3. Conclusion

In this paper, we propose a machine learning (ML) model metadata format to enhance the FAIRness of ML models, addressing key challenges in adhering to FAIR principles, particularly in sharing datasets, source code, and the ML model metadata. We expect this format to evolve based on user feedback and the rapidly changing needs of the ML field, which is shaping the future of artificial intelligence. Additionally, we offer insights into the development of a layered architecture for ML project metadata, focusing on distinct aspects of evaluating FAIRness. The format also introduces primitives for linking the ML project metadata across existing vocabularies, fostering interoperability. We plan to extend the MLDCAT-AP [9] by incorporating the metadata outlined here to support the representation of ML projects on platforms like Hugging Face [10].

[9] MLDCAT-AP: https://semiceu.github.io/MLDCAT-AP/releases/2.0.0/
[10] HuggingFace Models: https://huggingface.co/models

Acknowledgement

This work has been funded by the German Research Foundation (DFG) under project number 460234259 (NFDI4DataScience).

References

[1] T. A. Taffa, R. Usbeck, Leveraging llms in scholarly knowledge graph question answering., in: QALD/SemREC@ ISWC, 2023.

[2] S. Hertling, H. Paulheim, Olala: Ontology matching with large language models, in: Proceedings of the 12th Knowledge Capture Conference 2023, 2023, pp. 131–139.

[3] S. Bianco, L. Celona, M. Donzella, P. Napoletano, Improving image captioning descriptiveness by ranking and llm-based fusion, arXiv preprint arXiv:2306.11593 (2023).

[4] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons, The FAIR Guiding Principles for scientific data management and stewardship, Scientific Data 3 (2016) 160018. URL: https://www.nature.com/articles/sdata201618. doi:10.1038/sdata.2016.18.

[5] M. Akhtar, O. Benjelloun, C. Conforti, P. Gijsbers, J. Giner-Miguelez, N. Jain, M. Kuchnik, Q. Lhoest, P. Marcenac, M. Maskey, et al., Croissant: A metadata format for ml-ready datasets, in: Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning, 2024, pp. 1–6.

[6] M. L. B. Joaquin Vanschoren, C. S. Ong, Open Science in Machine Learning, Implementing Reproducible Research, 2014.

[7] S. Raza, S. Ghuge, C. Ding, E. Dolatabadi, D. Pandya, Fair enough: How can we develop and assess a fair-compliant dataset for large language models' training?, 2024. URL: https://arxiv.org/abs/2401.11033. arXiv:2401.11033.

[8] O. E. Gundersen, S. Kjensmo, State of the art: Reproducibility in artificial intelligence, Proceedings of the AAAI Conference on Artificial Intelligence 32 (2018). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11503. doi:10.1609/aaai.v32i1.11503.

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

[10] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu,

X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: https://www.tensorflow.org/, software available from tensorflow.org.

[11] B. Wentzel, F. Kirstein, T. Jastrow, R. Sturm, M. Peters, S. Schimmler, An extensive methodology and framework for quality assessment of dcat-ap datasets, in: I. Lindgren, C. Csáki, E. Kalampokis, M. Janssen, G. Viale Pereira, S. Virkar, E. Tambouris, A. Zuiderwijk (Eds.), Electronic Government, Springer Nature Switzerland, Cham, 2023, pp. 262–278.

Appendices

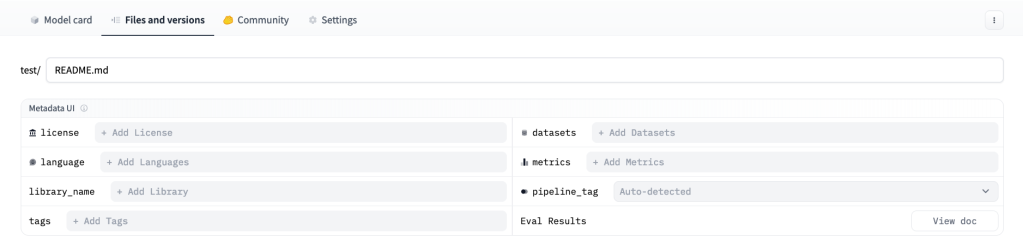
Metadata for Models

Table 1
Metadata from *SoftwareSourceCode* and *SoftwareApplication* of Schema.org.

Criteria	Metadata
Findability	codeRepository
Accessibility	codeRepository, url, acquireLicensePage, license
Interoperability	accessibilityAPI
Reusability	codeRepository, programmingLanguage, runtimePlatform, softwareRequirements, memoryRequirements

Table 2
Illustration of semantically similar metadata between Hugging Face Model Cards and Schema.org

Model Card (Hugging Face)	Metadata (schema.org)	Description
license	*license*	any valid license identifier
language	*inLanguage*	list of ISO 639-1 code for your language
library_name	*softwareRequirements*	used libraries in models
tags	*abstract* or *keywords*	domain used in models
datasets	-	datasets listed under models, e.g., imdb
metrics	-	evaluation metrics for a model, along with their values
pipeline_tags	-	indicating the type of task the model is intended for, e.g., Question Answering
Eval Results	-	model's evaluation results



test/ README.md

license	datasets
language	metrics
library_name	pipeline_tag: Auto-detected
tags	Eval Results

Figure 1: Metadata User Interface of Hugging Face Model Card.