Macaroni: Crawling and Enriching Metadata from Public Model Zoos

Li, Z.; Hai, R.; Katsifodimos, A; Bozzon, A.

DOI
10.1007/978-3-031-34444-2_31

Publication date
2023

Published in
International Conference on Web Engineering

Macaroni: Crawling and Enriching Metadata from Public Model Zoos

Ziyu Li(✉), Henk Kant, Rihan Hai, Asterios Katsifodimos, and Alessandro Bozzon

Delft University of Technology, Delft, The Netherlands
Z.Li-14@tudelft.nl

Abstract. Machine learning (ML) researchers and practitioners are building repositories of pre-trained models, called *model zoos*. These model zoos contain metadata that detail various properties of the ML models and datasets, which are useful for reporting, auditing, reproducibility, and interpretability. Unfortunately, the existing metadata representations come with limited expressivity and lack of standardization. Meanwhile, an interoperable method to store and query model zoo metadata is missing. These two gaps hinder model search, reuse, comparison, and composition. In this demo paper, we advocate for standardized ML model metadata representation, proposing *Macaroni*, a metadata search engine with toolkits that support practitioners to obtain and enrich that metadata.

Keywords: Machine Learning · Model Zoo · Metadata Representation

1 Introduction

Organizations are creating collections of pre-trained machine learning (ML) models, also known as *model zoos* or *model repositories*, as an incentive for sharing and reusing ML models. The most widely used model zoos include HuggingFace, Tensorflow Hub, and PyTorch Hub[1]. For a model zoo, metadata is essential. The metadata describes necessary information about an ML model in the model zoo, including its inference classes, architecture, training dataset, hyperparameter configurations, evaluation performance, etc. Metadata reporting is developed in the context of an ongoing effort to promote Trustworthy and Responsible AI[2].

In this work, we reveal that by querying and enriching metadata for model zoos, it facilitates MLOps and opens up new opportunities for abundant use cases. Practical use cases include but not limited to: i) model discovery and model selection for downstream tasks [2]; ii) (semi-)automatic model composition for complex analytic tasks [7]; iii) optimizing ML workloads on heterogeneous infrastructure [6]; iv) model reproducibility and documentation; v) and AutoML [5].

[1] https://huggingface.co/, https://pytorch.org/hub/, www.tensorflow.org/hub.
[2] https://partnershiponai.org/paper/responsible-publication-recommendations/.

I. Garrigós et al. (Eds.): ICWE 2023, LNCS 13893, pp. 376–380, 2023.
https://doi.org/10.1007/978-3-031-34444-2_31

However, to fully explore the potential of model zoos for these use cases is non-trivial. A few challenges emerge due to the lack of structured and comprehensive metadata in existing solutions. First, aforementioned model zoos store various information about models, often in the form of model cards [4] with text fields primarily for human consumption. Model cards are not queryable, making it hard for automatic extension or management. Second, common ML model management tools/platforms, such as Amazon SageMaker, AzureML, MLflow, offer toolkits for practitioners to log metadata. However, logging the metadata is often optional, resulting in missing information and making it difficult for model discovery and model comparison. Moreover, existing model zoos often lack the support of fine-grained metadata. For instance, to select and fine-tune models for a specific task, e.g., sentiment analysis, an ML practitioner needs information regarding model definition, dataset statistics and the model's performance on the previous task. Such information is most times missing from model zoos such as HuggingFace. To summarize, a novel system that organizes the model zoo metadata in a *structured*, and *comprehensive* manner, is needed.

This demo paper represents an initial stride towards achieving our objective, wherein we introduce, *Macaroni*, a metadata search system for model zoos[3], along with a metadata model to represent ML models and related entities. Our system serves as a platform for querying and visualizing extensive metadata gathered from external model zoos or via offline automatic evaluations using easily-accessible APIs. Furthermore, Macaroni facilitates supplementing model cards and other Deep Learning benchmarks [1] by incorporating a wider range of evaluation metrics.

2 System Design

In this demo, we introduce the system architecture and the metadata model that represents different entities and relations. We also discuss the underlying technology components that obtain and enrich the metadata of model zoos. In particular, the main components include the metadata model, metadata crawler, and performance evaluation pipeline (on raw or perturbed datasets).

System Architecture. The system aims to query and enrich the metadata for model zoos. We present the system architecture in Fig. 1. The system includes a web-based interface as front-end and back-end with storage and computation. We collect metadata in two ways: crawling ML-related metadata from external model zoos; and enriching metadata, e.g., obtaining model performance by evaluating the model on raw or perturbed dataset. The metadata obtained is later stored in metadata storage supported by a metadata model. Macaroni allows users to i) interactively search/discover models, ii) compare multiple models on various objectives (e.g., accuracy, runtime, size), iii) and, specially, measure the robustness of models on perturbed dataset. These functionalities are novel compared to public model zoos and other metadata management tools, which can

[3] The prototype is available at metadatazoo.io.

be adopted seamlessly to support research studies such as explainable AI and AutoML.

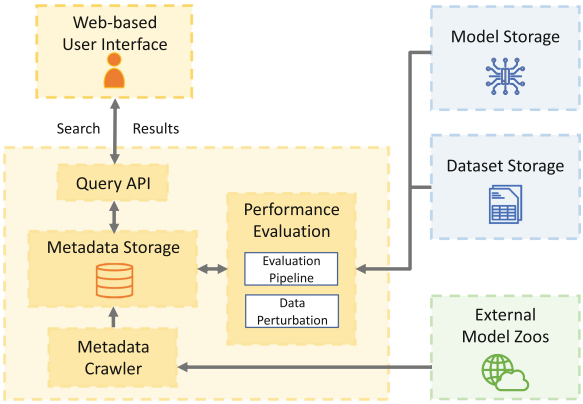

Fig. 1. The system architecture of Macaroni

Metadata Model. We present a conceptual view of our proposed metadata model [3], described using UML class diagrams. The metadata model is implemented with MangoDB (users can implement it in other database. The metadata model is comprised of four packages: i) `Configuration` package, which defines the ML models associated with architecture, hyperparameter settings, input and output. ii) `Dataset` packaged, includes both `Dataset`s and their *Data Instance*s. With the `Dataset` element, we represent the metadata of the datasets that is significant for data management and reporting, e.g., data source, data version. iii) `Execution` package describes the inference results (e.g., inference classes) of the model, possibly enriched with description from a knowledge graph. iv) `Evaluation` package, which presents the run-time metrics obtained in specified hardware settings. The instances of the `Evaluation` includes various types of evaluation metrics.

Metadata Crawler. The tool can automatically extract information from external model zoos, including HuggingFace and FiftyOne[4]. Metadata can be extracted from their APIs or information on the web pages and is recorded along with the source in the metadata storage. In particular, the model name, hyperparameters and task are obtained from the model cards from both model zoos through their APIs, and stored alongside the origin of the model. Information not available through their API (e.g., datasets and other tags) is parsed from the model's original readme files shown as the model cards. Other metadata could also be retrieved depending on the content provided by the external model zoos. Future work can investigate on extracting knowledge as metadata

[4] https://docs.voxel51.com/.

from the textual descriptions. Since more models will be added/updated , the crawling and extraction of the metadata shall be updated from time to time.

Model Evaluation Pipeline. The execution of models from different model zoos can be varied, differing by framework, algorithm, tasks, training dataset(s), and input format. To obtain the model performance and compare the evaluation results, we provide a unified evaluation pipeline, which facilitates evaluating models from different external model zoos on various datasets conveniently. Our pipeline is extensible, i.e., add support for new types of models or data after the initial pipeline deployment. We achieve extensibility in two ways. i) We apply a modular design, in which each evaluation module defines how to evaluate for a subset of models. ii) From each evaluation module, one or more evaluations are conducted, based on configuration of the module, such as which metrics to calculate or datasets to use.

Data Perturbation to Measure Model Capabilities. In addition to model performance on dedicated dataset, we also allow practitioners to investigate the model robustness on various types of data changes. Identifying the model performance on different data shifts is fundamental in understanding the model capabilities. We define a few types of perturbations on input dataset, e.g., converting to greyscale, flipping or mirroring the images, and observe the difference in performance. In such a way, we manage to identify how the model is generalized to data with different properties/changes. Future work can incorporate different perturbation methods, e.g., adversarial attacks, adding noise, on various modalities. We view the establishment of such a way to observe the model capabilities as the starting step, and further techniques and methods from explainable AI can enrich the description for model capabilities.

3 Conclusion and Future Work

We advocate for the need of structured, queryable, and comprehensive metadata for model zoos. We have proposed a platform for managing ML-related metadata including interactive web-based interface and related functionalities. To obtain the performance of models with various configurations, we also provide easily accessible APIs for model inference and evaluation. Our proposed platform can be used for building systems of AutoML, complex analytic tasks with model composition and explainable AI.

References

1. Coleman, C.: Dawnbench: an end-to-end deep learning benchmark and competition. Training **100**(101), 102 (2017)
2. Deshpande, A., et al.: A linearized framework and a new benchmark for model selection for fine-tuning. arXiv preprint arXiv:2102.00084 (2021)
3. Li, Z., Hai, R., et al.: Metadata representations for queryable ML model zoos. https://doi.org/10.48550/ARXIV.2207.09315, https://arxiv.org/abs/2207.09315

4. Mitchell, M., et al.: Model cards for model reporting. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 220–229 (2019)
5. Vanschoren, J.: Meta-learning. In: Automated Machine Learning: Methods, Systems, Challenges, pp. 35–61 (2019)
6. Wu, Y., Lentz, M., Zhuo, D., Lu, Y.: Serving and optimizing machine learning workflows on heterogeneous infrastructures
7. Yang, Z., et al.: Optimizing machine learning inference queries with correlative proxy models. arXiv preprint arXiv:2201.00309 (2022)