MRM3: Machine Readable ML Model Metadata

Andrej Čop, Blaž Bertalanič, Marko Grobelnik, Carolina Fortuna Jožef Stefan Institute Ljubljana, Slovenia ac8560@student.uni-lj.si

Abstract

As the complexity and number of machine learning (ML) models grows, well-documented ML models are essential for developers and companies to use or adapt them to their specific use cases. Model metadata, already present in unstructured format as model cards in online repositories such as Hugging Face, could be more structured and machine readable while also incorporating environmental impact metrics such as energy consumption and carbon footprint. Our work extends the existing State of the Art by defining a structured schema for ML model metadata focusing on machine-readable format and support for integration into a knowledge graph (KG) for better organization and querying, enabling a wider set of use cases. Furthermore, we present an example wireless localization model metadata dataset consisting of 22 models trained on 4 datasets, integrated into a Neo4j-based KG with 113 nodes and 199 relations.

Keywords

model metadata, knowledge graph, neo4j, ontology, taxonomy, machine learning

ACM Reference Format:

Andrej Čop, Blaž Bertalanič, Marko Grobelnik, Carolina Fortuna. 2025. MRM3: Machine Readable ML Model Metadata. In 3rd International Workshop on Networked AI Systems (NetAISys '25), June 27, 2025, Anaheim, California, United States. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3711875.3736685

1 Introduction

ML models have become essential for modern intelligent systems, enabling automation and data-driven decision making across numerous domains such as modern 5G/6G networks [16] or Internet of Things [7]. As the complexity and diversity of ML models grow, systematic representation and organization of model and dataset metadata become increasingly important. Model metadata containing information describing a model's development context and characteristics such as training process, performance, bias, and input datasets is the foundation for informed decisions regarding model selection, deployment, retraining, and optimization [21].

Model repositories such as Hugging Face (HF)¹, focus on providing model and dataset cards, which consist of unstructured metadata

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

NetAISys'25 , Anaheim, California, $United\ States$

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-1453-5/25/06

https://doi.org/10.1145/3711875.3736685

intended primarily for human readability [17]. Model and dataset cards consists of some standardized representation, and existing work attempts to define what to include in the documentation [19]. Specifically, for ML dataset cards, there is existing work on machine readability. Structured representation and standard format of dataset cards allow not only human readability but also enable intelligent searching for the appropriate datasets by automated means, therefore accelerating the creation of new models on existing, well-documented datasets [2, 10, 11].

Current approaches to model documentation, such as model cards, provide valuable descriptive information but typically exist as unstructured text, making systematic comparison and selection challenging. Although significant progress has been made in dataset documentation through frameworks for ML datasets, such as DescribeML [10] and Croissant [2], equivalent structured representations for model metadata remain underdeveloped. The absence of a standardized machine-readable representation of model metadata inhibits the development of fully automated intelligent systems for model orchestration and selection.

Sustainability metrics such as carbon footprint and energy consumption are becoming important as the environmental impact of ML model training and inference is explored [24]. These metrics should be consistently included in such documentation allowing direct comparison between different models and enabling the selection of the best ML model based on multiple criteria, such as performance metrics, computational requirements, and environmental impact, which are specifically useful in low-powered network edge devices. To achieve this, a novel approach to representation of structured model metadata is needed. A graph database that connects different entities with relations, improves the ability to select related ML models based on various criteria, as different ML models can be highly correlated. For example, models trained on the same dataset with different architecture or parameters or the models for same service. This paper addresses these limitations by introducing a solution for representing ML model metadata through a KG approach. Our contributions are as follows.

- We identify a taxonomy of ML model metadata and propose an ontology defining the relationships between model metadata entities, enabling structured representation in a KG
- Based on proposed ontology we define a Neo4j based Machine Readable ML Model Metadata (MRM3) knowledge graph.
 We show an example KG using model metadata from 22 trained wireless localization models. The localization MRM3 consists of 113 nodes and 199 relations and enables querying of models by lowest environmental impact while satisfying the required accuracy.
- By open-sourcing our code, providing an example localization model metadata dataset, and offering a standardized

¹https://huggingface.co/

JavaScript Object Notation (json) schema for collecting and representing model metadata, we enable and simplify the implementation of the proposed MRM3 for any user or organization.

The remainder of this paper is organized as follows. In Section 2 we discuss the related work, while in Section 3 we identify ML model metadata taxonomy and propose ontology for MRM3. In Section 4 we present an example localization MRM3. Finally, we conclude the paper in Section 5.

2 Related Work

Existing work on ML dataset and model documentation has highlighted the need for more structured and precise metadata representation. Giner-Miguelez et al. [10, 11] proposed a domain-specific language (DSL) and a DescribeML tool to document ML dataset structure, provenance, and fairness. On the other hand, Akhtar et al. [2] introduced Croissant, a metadata format adopted by HF, improving dataset discoverability and interoperability across the ML field. While we do not focus on structured dataset documentation this related work inspired machine readability of our proposed model metadata structure.

Pepe et al. [21] and Liang et al. [17] analyzed HF model cards, finding limited disclosure of datasets, biases, and licenses, with Liang et al. noting improved downloads with more detailed cards. Similarly, Yang et al. [25] performed an analysis of dataset cards on HF platform. Model and dataset cards on HF are in a human-readable format and provide useful documentation for developers but lack a consistent schema and ontology, which we address in this work.

Mitchell et al. [19] proposed model cards for transparent reporting of ML model performance across diverse conditions at Google. Beyond the research paper, Google has developed a suite of tools for structured model documentation, including the Model Card Toolkit [23], which provides a json schema for standardizing model reporting. Their ML Metadata (MLMD) library further enables tracking and managing model metadata throughout the ML lifecycle [12]. While Google's json schema standardizes model reporting, our work extends this by defining a structured schema for ML model metadata focusing on machine-readable format and support for integration into a KG for better organization and querying, enabling a wider set of use cases.

McMillan-Major et al. [18] developed reusable natural language processing documentation templates for HF. On the other hand, Bhat et al. [5] and Crisan et al. [6] addressed documentation gaps, introducing DocML for ethical reporting and interactive model cards to aid non-expert understanding, respectively. Amith et al. [3] formalized model cards into an OWL2 ontology for machine-readable bioinformatics models, while Ajibode et al. [1] critiqued inconsistent versioning in HF language models, recommending semantic standards.

In contrast to these efforts on dataset and model documentation, our work uniquely integrates a taxonomy, ontology, and Neo4j-based KG for ML model metadata.

3 ML Model Metadata Taxonomy and Ontology

Documentation of datasets and ML models enables users to understand and effectively utilize published open-sourced ML models in their applications. An example of such documentation is presented as dataset and model cards on a popular model repository platform HF. Based on analyses of existing work and proposed approaches to ML model documentation we can identify a common structure and taxonomy for such documentation, focusing on attributes that are important for the automated selection of ML models.

In this section, we define the taxonomy of ML model metadata and using an established methodology for KG development [22] we can construct an ontology, and develop a KG for MRM3.

3.1 Identified Taxonomy

Based on prior work on model card structure [3, 19], analysis of model cards on HF model repository [17, 18] and inspired by the previously proposed structure of machine-readable dataset metadata [2, 10, 11] we identify taxonomy of ML model metadata. This taxonomy provides a structure on what specific metadata to extract when training and testing ML models. The identified taxonomy is presented in Fig. 1 and further explained in subsequent paragraphs.

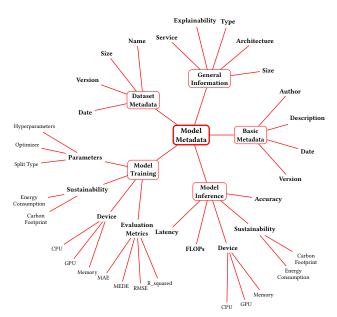


Figure 1: Taxonomy of ML model metadata.

The main branches of model metadata are model training, model inference, dataset metadata, general information and basic metadata as depicted in Fig. 1.

Model training metadata refers to information about how and where the model was trained and which evaluation metrics are collected. For example, model training parameters and hyperparameters and evaluation metrics such as root mean squared error (RMSE) or mean absolute error (MAE). However, it is not limited to metrics for regression but can be extended to include evaluation metrics for tasks such as clustering and classification. Further, device metadata describes the CPU, GPU, and memory information of

the device on which the model was trained. We identified sustainability as an important part of model training [24]. Sustainability is further divided into energy consumption and carbon footprint as can be seen in the left bottom part of Fig. 1.

Model inference metadata, similarly to training metadata, includes sustainability metrics and information about the device on which the inference was performed. It also provides end-to-end latency information for inference requests and computational complexity which refers to the number of Floating Point Operations (FLOPs) required for running inference on a single input sample [8].

Dataset metadata, presented as *Dataset* in Fig. 1, includes essential information to uniquely identify datasets such as date, version, size and name, following the commonly found dataset metadata proposed in DescribeML framework [10] and Croissant metadata format [2].

Similarly, general model information and basic metadata include information about date, version, and size as well as model architecture, the service for which the model was intended, and the architecture's explainability. It can also include information such as a description of the models and a list of authors. General model information is depicted in the top right of the ML model metadata taxonomy and basic metadata on the right side in Fig. 1.

While information such as license, citation details, and associated research papers are common in model cards [18, 19] we don't identify them as important in our model metadata taxonomy as they are not crucial for use cases such as automated selection of ML models. However, the taxonomy can be extended to also include such information.

3.2 Ontology

Taxonomy of ML model metadata, identified in Section 3.1, helps developers decide on which metadata to collect and store during the process of development, training, and testing of ML models, however, this metadata by itself does not support connecting different ML models and creating relations between them. For this reason, we define ontology [13] of ML model metadata. An ontology defines the structure of the KG and allows a model of how the KG is represented in a structured way with relations between concepts [22].

Using the identified taxonomy in Section 3.1 and presented in Fig. 1, we propose an ontology of ML model metadata presented in Fig. 2. The ontology is proposed based on relations between definitions in the taxonomy and using the methodology for creating knowledge graphs [22].

As we can see from Fig. 2, general information about the model is contained in the main node, called *Model*. A relation *TRAINED_ON* connecting node *Model* to node *Dataset* defines the dataset that the model was trained on including dataset metadata. The relation *PROVIDES* connects models to their respective service and its requirements for example localization service, text generation, and image classification. *Service* node connects to *ProblemType* with relation *SOLUTION_FOR*, which defines the type of ML problem such as regression, classification, clustering etc. The relation *UTILIZES* and node *ModelArchitecture* define the architecture of the ML model.

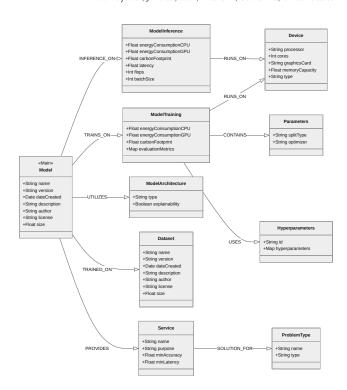


Figure 2: Ontology of ML model metadata.

In Fig. 2 we can see relation <code>TRAINS_ON</code> which connects node <code>Model</code> to node <code>ModelTraining</code>. <code>ModelTraining</code> includes attributes concerning training of the ML model and itself connects to node <code>Parameters</code>, and <code>Hyperparameters</code> which define a set of parameters and hyperparameters that the model is trained with. Similarly, node <code>ModelInference</code> describes attributes concerning ML models in prediction mode, such as latency and FLOPs. As we can see in the bottom left of Fig. 2, both <code>ModelTraining</code> and <code>ModelInference</code> connect to <code>Device</code> with relation <code>RUNS_ON</code> which specify the device and its hardware that the model was trained on or performed inference.

3.3 Knowledge Graph

Knowledge graphs are graph-structure knowledge bases [22]. They use a graph data model to capture knowledge, providing an abstraction of domains, while capturing relations between entities in these domains. Ontologies define the structure of knowledge graphs, while queries enable clustering, summarization, and insight generation about the described domain [15].

To create a knowledge graph that can support use cases such as automated selection of ML models, a methodology of identifying data, constructing ontology, extracting knowledge, processing knowledge, and constructing the knowledge graph was used [22]. Based on the constructed ontology in Section 3.2 and Fig. 2 we can create KG for any dataset of model metadata that follows the identified taxonomy of model metadata in Section 3.1 and Fig. 1.

We utilize Neo4j Graph Database Platform² to support the construction of our knowledge graph and enable queries using Cypher

²https://neo4j.com/

language. Neo4j is one of the top open-source graph database [14] and is widely used in the knowledge graph development process [22].

4 Localization MRM3

In this section, we present an example MRM3 dataset and knowledge graph. The localization model metadata dataset³ is based on model metadata collected from wireless localization models of different architectures and on different wireless datasets, namely Lumos5G [20], LOG-a-TEC [4] (winter and spring), and UMU^4 [9], using low-code-localization pipeline. Using identified taxonomy in Section 3.1 we collected the required metadata and inserted it into the Neo4j-based knowledge graph with the structure as defined in Section 3.2.

4.1 Json Schema

To have a consistent way of collection and structure of metadata we propose a model metadata json schema. Which is available on MRM3 Github repository⁵ and can be used to verify the structure of any json ML model metadata dataset. It defines which metrics to collect, how to structure them in json format. The schema includes short descriptions of metrics and in which units of measurement they should be stored.

4.2 Model Metadata Knowledge Graph

From the localization training pipeline, we collected json model metadata following the proposed json schema in the preceding section for 4 different datasets and multiple architectures such as random forest, k-nearest neighbors and XGBoost. We developed a python interface to insert any json file following the defined json schema into a deployed Neo4j KG instance⁵.

Table 1 presents the number of nodes and entities in our localization KG. Our example KG consists of 22 models trained on 4 datasets with models based on 4 architectures. All the training and inference was performed on one device and in total 44 nodes have relation *RunsOn* to the device, as we can see from Table 1. In total it includes 113 nodes and 199 relations. The details of relations, nodes and which properties they contain are explained in Section 2 and in Fig. 2.

KG Entities	Quantity
All Relations	199
All Nodes	113
Relation RunsOn	44
Node <i>Model</i>	22
Node Dataset	4
Node ModelArchitecture	4
Node <i>Device</i>	1

Table 1: Summary of localization MRM3.

4.3 Use Case

Wireless localization MRM3 enables use cases such as the selection of ML models and providing insight into ML models and their metadata. Specifically for network edge needs where efficient computation of model inference is crucial, MRM3 enables intelligent selection of ML models based on certain search criteria such as low computational complexity for inference or minimizing energy consumption.

By integrating MRM3 to Machine Learning Operations (MLOps) systems, which orchestrate the process of ML training and deployment on distributed and edge infrastructure, this use case improves the process of orchestrating ML models and could improve energy consumption and inference latency on low power edge devices [26].

4.4 Querying the Knowledge Graph

Cypher is a declarative language to query information from the Neo4j graph database [14]. It can be used to interact and query knowledge from any MRM3. Using Cypher it is possible to query and display all data results in a graph depicted in Fig. 3. It displays all nodes and relations in the localization MRM3 with one property written in the center of each node. Each type of node is colored by different color, for example, dataset nodes are colored orange and there are four of them as we can confirm in Table 1 summarizing the localization MRM3.

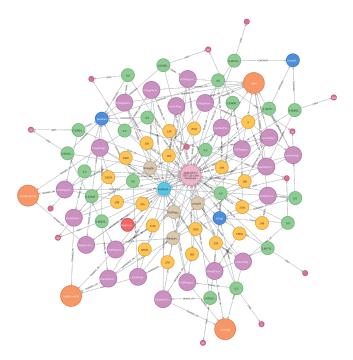


Figure 3: Localization MRM3 on Neo4j.

To enable the use case described in Section 4.3 we provide an example Cypher query in Listing 1. The query returns models ordered by lowest energy consumption for model inference, while also providing computational complexity for inference and architecture of the model.

³https://doi.org/10.1145/3711875.3736685

 $^{^4\}mathrm{The}$ data used in this study was collected as part of the EU Nancy project and it will be publicly available soon.

⁵https://github.com/sensorlab/MRM3

As we can see in Listing 1, it first matches *Model* nodes that have associated *Dataset* and *ModelArchitecture* information in the KG. The query connects to *ModelInference* nodes through the *INFERENCE_ON* relationships to extract inference metrics about energy consumption and computational complexity. The results are ordered by ascending energy consumption values in the last line in Listing 1.

Listing 1: Example code to query Neo4j based MRM3 using Cypher.

The query output by Listing 1 of the top five models is presented in Table 2. The first column describes the model architecture, while the second specifies the dataset used for training. The third column lists the energy consumption per inference request (in joules), sorted in ascending order. The final column indicates the computational complexity measured in FLOPs.

Architecture	Dataset	Energy (J)	FLOPs
Random Forest	UMU	0.072	249
Random Forest	Lumos5G	0.132	263
XGBoost	LOG-a-TEC Winter	0.284	140
KNeighbors	UMU	0.326	134
Random Forest	LOG-a-TEC Spring	0.370	246

Table 2: Output of example query in Listing 1.

Executing the query in Listing 1 takes approximately 3 ms on average, while retrieving all data shown in Fig. 3 takes about 7 ms on average.

4.5 Implementation Considerations

To construct the model metadata KG, the json schema proposed in Section 4.1 serves as a template to determine which data should be collected from the model training project, how it should be structured in json files, and in which units of measurement the collected metrics should be stored. Training code or training pipeline must be modified to include measurements of required metrics, while static information such as authors, dataset metadata, and device properties should be collected and inserted into the dataset.

Such a dataset can be inserted in a Neo4j instance in the same manner as our example localization MRM3, discussed in Section 4.2 while the entire process is documented in MRM3 Github repository⁵. The practical implementation requires deploying a Neo4j graph database instance, which can be accomplished using Docker containerization. Prior to knowledge graph construction, json files

should be validated against the model card schema using the provided validation utility to ensure the correct structure.

The transformation process from json metadata to graph entities involves creation of nodes and relationship while following the structure defined in the ontology.

Once deployed, this domain-specific metadata knowledge graph enables use cases including automated model selection based on specific constraints such as energy consumption as presented in Listing 1 in Section 4.4.

As new ML models are trained and new metadata information is collected, this data can be inserted into the deployed Neo4j instance following the same process outlined in the preceding paragraphs to maintain consistent ontology structure. The transformation from json files to graph entities remains unchanged. This continuous update of the knowledge graph enables selection of newly added models across an expanding collection of ML model metadata.

5 Conclusion

This paper presents a solution for structured representation of ML model metadata through a knowledge graph approach. Our proposed *Machine Readable ML Model Metadata* addresses the critical gap in machine-readable model documentation by establishing a standardized ontology that systematically captures relationships between models, datasets, architectures, and performance metrics. By implementing this ontology in Neo4j and showing an example localization MRM3 with 22 localization models trained on 4 datasets, we demonstrated the practical utility of our approach, enabling efficient querying of models based on multiple criteria including environmental impact metrics. The standardized json schema we developed enables consistent metadata collection across diverse ML projects, simplifying integration with existing workflows.

In our future work, we will show and implement the discussed use cases, such as automated selection of ML models for the example localization MRM3 and extend the domain to ML models beyond network systems.

Acknowledgments

This work was supported by the Slovenian Research Agency (P2-0016) and the European Commission NANCY project (No. 101096456). We would like to acknowledge the SensorLab team for developing the localization models, and Gregor Cerar and Tim Strnad for developing an automated training pipeline.

References

- Adekunle Ajibode, Abdul Ali Bangash, Filipe Roseiro Cogo, Bram Adams, and Ahmed E. Hassan. 2025. Towards Semantic Versioning of Open Pre-trained Language Model Releases on Hugging Face. arXiv:2409.10472 [cs.SE] https://arxiv.org/abs/2409.10472
- [2] Mubashara Akhtar, Omar Benjelloun, Costanza Conforti, Pieter Gijsbers, Joan Giner-Miguelez, Nitisha Jain, Michael Kuchnik, Quentin Lhoest, Pierre Marcenac, Manil Maskey, Peter Mattson, Luis Oala, Pierre Ruyssen, Rajat Shinde, Elena Simperl, Goeffry Thomas, Slava Tykhonov, Joaquin Vanschoren, Jos van der Velde, Steffen Vogler, and Carole-Jean Wu. 2024. Croissant: A Metadata Format for ML-Ready Datasets. In Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning (Santiago, AA, Chile) (DEEM '24). Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/3650203.3663326
- [3] Muhammad Tuan Amith, Licong Cui, Degui Zhi, Kirk Roberts, Xiaoqian Jiang, Fang Li, Evan Yu, and Cui Tao. 2022. Toward a standard formal semantic representation of the model card report. *BMC Bioinformatics* 23 (2022). doi:10.1186/s12859-022-04797-6

- [4] Blaž Bertalanič, Grega Morano, and Gregor Cerar. 2022. Lokalizacija na prostem s pomočjo radiofrekvenčnih odtisov tehnologije BLE v testnem omrežjem LOG-a-TEC = LOG-a-TEC Testbed outdoor fingerprinting localization using BLE beacons. Slovenska sekcija IEEE; Fakulteta za elektrotehniko, 50–53. https://erk.fe.unilj.si/2022/erk22.pdf Nasl. z nasl. zaslona.
- [5] Avinash Bhat, Austin Coursey, Grace Hu, Sixian Li, Nadia Nahar, Shurui Zhou, Christian Kästner, and Jin L.C. Guo. 2023. Aspirations and Practice of ML Model Documentation: Moving the Needle with Nudging and Traceability. In Conference on Human Factors in Computing Systems - Proceedings. doi:10.1145/3544548. 3581518
- [6] Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. 2022. Interactive Model Cards: A Human-Centered Approach to Model Documentation. In ACM International Conference Proceeding Series. doi:10.1145/3531146.3533108
- [7] Laizhong Cui, Shu Yang, Fei Chen, Zhong Ming, Nan Lu, and Jing Qin. 2018. A survey on application of machine learning for Internet of Things. *International Journal of Machine Learning and Cybernetics* 9, 8 (2018). doi:10.1007/s13042-018-0834-5
- [8] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. 2023. Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. Sustainable Computing: Informatics and Systems 38 (2023). doi:10.1016/j.suscom.2023.100857
- [9] Jorge Gallego-Madrid, Luis Bernal-Escobedo, Rodrigo Asensio, Ana Hermosilla, Alejandro Molina Zarca, Jordi Ortiz, Ramon Sanchez-Iborra, and Antonio Skarmeta. 2022. GAIA 5G: A Multi-access Smart-Campus Architecture. In Internet of Things, Aurora González-Vidal, Ahmed Mohamed Abdelgawad, Essaid Sabir, Sébastien Ziegler, and Latif Ladid (Eds.). Springer International Publishing, Cham, 363–374.
- [10] Joan Giner-Miguelez, Abel Gómez, and Jordi Cabot. 2022. DescribeML: a tool for describing machine learning datasets. In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings (Montreal, Quebec, Canada) (MODELS '22). Association for Computing Machinery, New York, NY, USA, 22–26. doi:10.1145/3550356.3559085.
- [11] Joan Giner-Miguelez, Abel Gómez, and Jordi Cabot. 2023. A domain-specific language for describing machine learning datasets. *Journal of Computer Languages* 76 (2023). doi:10.1016/j.cola.2023.101209
- [12] Google. 2023. ML Metadata (MLMD). https://github.com/google/ml-metadata Accessed: 2025-03-05.
- [13] Nicola Guarino, Daniel Oberle, and Steffen Staab. 2009. What Is an Ontology? In Handbook on Ontologies. doi:10.1007/978-3-540-92673-3{_}}0
- [14] José Guia, Valéria Gonçalves Soares, and Jorge Bernardino. 2017. Graph databases: Neo4j Analysis. In ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Vol. 1. doi:10.5220/0006356003510356
- [15] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. Comput. Surveys 54, 4 (2021). doi:10.1145/3447772
- [16] Khaled B. Letaief, Wei Chen, Yuanming Shi, Jun Zhang, and Ying Jun Angela Zhang. 2019. The Roadmap to 6G: AI Empowered Wireless Networks. IEEE Communications Magazine 57, 8 (2019). doi:10.1109/MCOM.2019.1900271
- [17] Weixin Liang, Nazneen Rajani, Xinyu Yang, Ezinwanne Ozoani, Eric Wu, Yiqun Chen, Daniel Smith, and James Zou. 2024. Systematic analysis of 32,111 AI model cards characterizes documentation practice in AI. Nature Machine Intelligence 6 (06 2024), 1–10. doi:10.1038/s42256-024-00857-z
- [18] Angelina McMillan-Major, Salomey Osei, Juan Diego Rodriguez, Pawan Sasanka Ammanamanchi, Sebastian Gehrmann, and Yacine Jernite. 2021. Reusable Templates and Guides For Documenting Datasets and Models for Natural Language Processing and Generation A Case Study of the HuggingFace and GEM Data and Model Cards. In GEM 2021 - 1st Workshop on Natural Language Generation, Evaluation, and Metrics, Proceedings. doi:10.18653/v1/2021.gem-1.11
- [19] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency. doi:10.1145/3287560.3287596
- [20] Arvind Narayanan, Eman Ramadan, Rishabh Mehta, Xinyue Hu, Qingxu Liu, Rostand A. K. Fezeu, Udhaya Kumar Dayalan, Saurabh Verma, Peiqi Ji, Tao Li, Feng Qian, and Zhi-Li Zhang. 2020. Lumos5G: Mapping and Predicting Commercial mmWave 5G Throughput. In Proc. ACM Internet Measurement Conference (Virtual Event, USA) (IMC '20). Association for Computing Machinery, New York, NY, USA, 176-193. doi:10.1145/3419394.3423629
- [21] Federica Pepe, Vittoria Nardone, Antonio Mastropaolo, Gabriele Bavota, Gerardo Canfora, and Massimiliano Di Penta. 2024. How do Hugging Face Models Document Datasets, Bias, and Licenses? An Empirical Study. In Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension (Lisbon, Portugal) (ICPC '24). Association for Computing Machinery, New York, NY, USA, 370–381. doi:10.1145/3643916.3644412

- [22] Gyte Tamašauskaite and Paul Groth. 2023. Defining a Knowledge Graph Development Process Through a Systematic Review. ACM Transactions on Software Engineering and Methodology 32, 1 (2023). doi:10.1145/3522586
- [23] TensorFlow. 2023. Model Card Toolkit. https://github.com/tensorflow/modelcard-toolkit Accessed: 2025-03-05.
- [24] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. Proceedings of Machine Learning and Systems 4 (2022), 795–813.
- [25] Xinyu Yang, Weixin Liang, and James Zou. 2024. Navigating Dataset Documentations in AI: A Large-Scale Analysis of Dataset Cards on Hugging Face. arXiv:2401.13822 [cs.LG] https://arxiv.org/abs/2401.13822
- [26] Andrej Čop, Blaž Bertalanič, and Carolina Fortuna. 2025. An overview and solution for democratizing AI workflows at the network edge. *Journal of Network* and Computer Applications 239 (2025), 104180. doi:10.1016/j.jnca.2025.104180