Science of Computer Programming 231 (2024) 103009

Contents lists available at ScienceDirect

Science of Computer Programming

journal homepage: www.elsevier.com/locate/scico

Original software publication

ModelSet: A labelled dataset of software models for machine learning

José Antonio Hernández López [a,*], Javier Luis Cánovas Izquierdo [b,*], Jesús Sánchez Cuadrado [a,*]

[a] Universidad de Murcia, Spain
[b] IN3 – UOC, Spain

ARTICLE INFO

Article history:
Received 7 December 2022
Received in revised form 18 July 2023
Accepted 7 August 2023
Available online 14 August 2023

Keywords:
Dataset
Software models
Model-driven engineering
Machine learning

ABSTRACT

Curated collections of models are essential for the success of Machine Learning (ML) and Data Analytics in Model-Driven Engineering (MDE). However, current datasets are either too small or not properly curated. In this paper, we present ModelSet, a dataset composed of 5,466 Ecore models and 5,120 UML models which have been manually labelled to support ML tasks. We describe the structure of the dataset and explain how to use the associated library to develop ML applications in Python. Finally, we present some applications which can be addressed using ModelSet.
Tool Website: https://github.com/modelset

Metadata

Table 1
Code metadata (mandatory).

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v1.0
C2	Permanent link to code/repository used for this code version	https://github.com/ScienceofComputerProgramming/SCICO-D-22-00335
C3	Permanent link to Reproducible Capsule	Not applicable
C4	Legal Code License	LGPL
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Python, Java
C7	Compilation requirements, operating environments and dependencies	Tested on Linux
C8	If available, link to developer documentation/manual	https://github.com/modelset
C9	Support email for questions	jesusc@um.es

1. Motivation and significance

The application of Machine Learning (ML) to solve Model-Driven Engineering (MDE) tasks has become an active research field. For instance, feed-forward neural networks are used to label software models [1], language models are used

* Corresponding authors.
E-mail addresses: joseantonio.hernandez6@um.es (J.A.H. López), jcanovasi@uoc.edu (J.L. Cánovas Izquierdo), jesusc@um.es (J.S. Cuadrado).

https://doi.org/10.1016/j.scico.2023.103009
0167-6423/

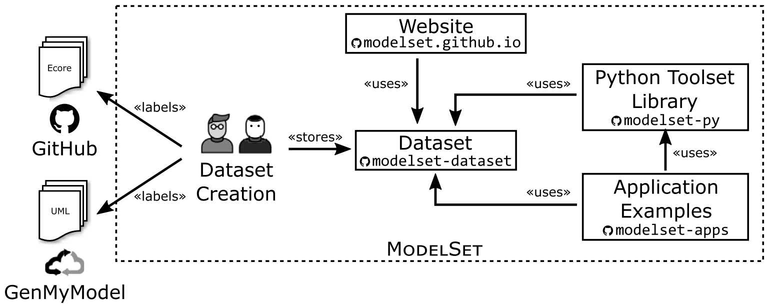

Fig. 1. Architecture of ModelSet.

to recommend modelling concepts [2], or graph neural networks are employed to assess how realistic model generators are [3].

Despite these efforts, one important shortcoming is the lack of large datasets [4], which severely limits the generalization of ML models. There are few datasets for MDE, and they are either too small or not curated. For instance, the labelled dataset of meta-models [5] only contains a few hundred of them, and G. Robles et al. [6] presented a dataset of 90 K models but only a few of them can be processed with standard frameworks like EMF. For the Business Process Modeling (BPM) domain, the BPM Academic Initiative made available a dataset of about 30 K BPMN models built with different modeling languages [7]. However, its curation process and the origin of the models is not fully specified (e.g., it contains some labels but does not seem to be systematically annotated, many models look synthetic, etc.).

To alleviate this situation, we created ModelSet, a dataset with more than 10 K models which have been manually labelled. While the labelling approach has been previously presented [4], in this paper we describe the ModelSet toolset, including its architecture, libraries and applications.

2. Software description

ModelSet is a dataset of software models which have been manually labelled and curated in order to foster research in ML and MDE. The models were extracted from an early snapshot of the models crawled by the MAR search engine[1] [8,9]. Then, they were labelled with their main category and several additional tags. The dataset contains 5,466 Ecore models and 5,120 UML models, and is provided with a set of Java and Python libraries to facilitate its use and illustrate its potential applications. Next sections will describe the architecture and the main functionalities.

2.1. Software architecture

The project is structured in four separate components, which have been developed as separated projects in the ModelSet Github's organization.[2] Fig. 1 shows the architecture of ModelSet.

Dataset. This component includes the relational databases and a set of Java-based tools to facilitate the release of new versions of the dataset. The project's name of this component is `modelset-dataset`.

Python Toolset Library. This library aims at facilitating loading ModelSet and using it to train ML algorithms in Python-based environments. The project's name of this component is `modelset-py`.

Application examples. This component contains illustrative examples using ModelSet to perform simple ML tasks with software models. The project's name of this component is `modelset-apps`.

Website. This component includes the code of the website of the tool, intended to explore the models and the labels of ModelSet. The project's name of this component is `modelset.github.io`.

2.2. Software functionalities

For users willing to use ModelSet, the software offer two main functionalities: (i) loading & exploring the dataset, and (ii) training ML models.

Loading & exploring the dataset. The dataset is provided as two relational databases for the Ecore and UML models, both in SQLite format. The database schema includes a table called *model* for basic model data (i.e., unique identifier, source repository and filename) and a table called *metadata* with label data (i.e., unique identifier and a JSON object with the label

[1] http://mar-search.org.
[2] https://github.com/modelset.

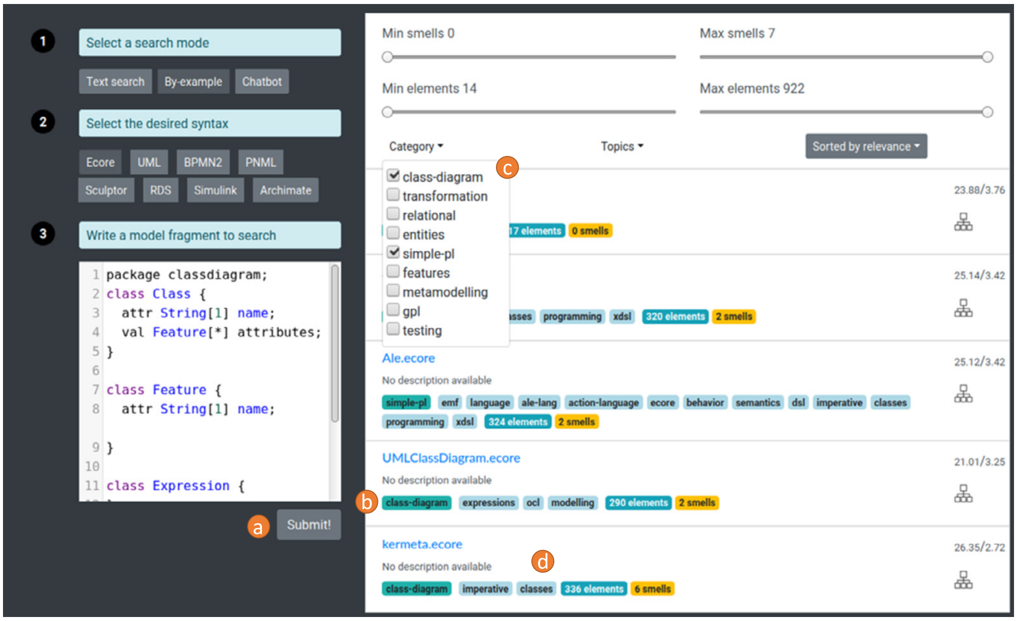

Fig. 2. Screenshot of MAR generating labels (from [4]).

information). The process of loading and accessing the databases is straightforward via SQLite plugins. Once the dataset is loaded, it can be easily explored with any database query interface using standard SQL queries.

Training ML models. The provided Python library is designed to facilitate the implementation of ML tasks with ModelSet. It helps to load the dataset and to prepare the data in a manner that is suitable for using well-known ML Python libraries (e.g., Scikit-Learn[3] or PyTorch[4]).

3. Illustrative examples

ModelSet has been previously used to address classification tasks for Ecore and UML models. These types of tasks can be informally defined as follows: given an unseen model, identify which is the most probable label (or labels if it is a multi-label task) for the model. Although this problem has been addressed before the release of ModelSet, it was done only with small datasets [10]. The usage of ModelSet allows us to derive stronger conclusions, as described in recent work [11].

Moreover, the classification model [4] has been used to implement a faceted search facility in the MAR search engine. Fig. 2 shows an example, in which the models retrieved as a response to a query are automatically provided a category (label b) and a set of tags (label d). It is also possible to refine the search using the categories (label c).

4. Impact

ModelSet has been recently published, and therefore its impact is still emerging, but we foresee a number of scenarios where the tool can be useful.

Evaluation of clustering methods. ModelSet gives access to numerous models which can be used to evaluate clustering methods. In this case, the labels can be interpreted as cluster identifiers and used as ground truth.

Recommender systems. ModelSet may be used when analyzing specific categories of models (e.g., a recommender system for the banking domain).

Spurious model identification. ModelSet may be used to identify spurious models, that is, low quality models (e.g., partial or wrong models).

Label-based stratified evaluation. ModelSet may be used to evaluate ML models using train-test-eval splitting in a stratified fashion using the labels. This is useful to avoid bias in performance estimation as there are several domains that contain much fewer models than others.

Train embeddings. ModelSet may be used to train embeddings of models of a domain (i.e., clone detection or recommender systems in a domain).

3 https://scikit-learn.org/stable/.
4 https://pytorch.org/.

Usage of models in education. ModelSet gives the opportunity to use models for illustrating purposes in educational environments.

Statistical model analysis. ModelSet size helps to apply several statistical analyses and perform quality assurance.

5. Conclusions

The application of ML algorithms to address tasks related to MDE is increasingly being researched. However, an important element which may hinder the evolution of this research line is the lack of sufficiently large datasets. In this paper, we have presented ModelSet, a large labelled dataset of software models composed of more than 10 K models, its architecture, main functionalities and illustrative examples of its usage.

6. Future plans

As future work, we plan to continue enhancing ModelSet. To this end, we aim at associating other types of labels, like textual descriptions of the models (e.g., to generate textual summaries). We also want to improve the Python library with new features like detection of duplicate models and better integration with other ML libraries. Finally, we are looking into ways to integrate the models provided by MAR with the Python library of ModelSet, to use them in unsupervised learning scenarios.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Work supported by grant TED2021-129381B-C22 (SATORI project) funded by MCIN/AEI/10.13039/501100011033 and NextGenerationEU/PRTR, and by grant PID2020-114615RBI00 (LOCOSS project) funded by MICIN/AEI/10.13039/501100011033.

Jesús Sánchez Cuadrado enjoys a grant RYC-2017-237 funded by MCIN/AEI/10.13039/501100011033 and by "ESF Investing in your future". José Antonio Hernández López enjoys a FPU grant funded by the Universidad de Murcia.

References

[1] P.T. Nguyen, J. Di Rocco, D. Di Ruscio, A. Pierantonio, L. Iovino, Automated classification of metamodel repositories: a machine learning approach, in: Int. Conf. on Model Driven Engineering Languages and Systems, 2019, pp. 272–282.

[2] M. Weyssow, H. Sahraoui, E. Syriani, Recommending metamodel concepts during modeling activities with pre-trained language models, Softw. Syst. Model. (2022) 1–19.

[3] J.A.H. López, J.S. Cuadrado, Towards the characterization of realistic model generators using graph neural networks, in: Int. Conf. on Model Driven Engineering Languages and Systems, 2021, pp. 58–69.

[4] J.A.H. López, J.L. Cánovas Izquierdo, J.S. Cuadrado, Modelset: a dataset for machine learning in model-driven engineering, Softw. Syst. Model. (2021) 1–20.

[5] Ö. Babur, A labeled ecore metamodel dataset for domain clustering, URL https://zenodo.org/record/2585456.

[6] G. Robles, T. Ho-Quang, R. Hebig, M.R. Chaudron, M.A. Fernandez, An extensive dataset of uml models in GitHub, in: Int. Conf. on Mining Software Repositories, IEEE, 2017, pp. 519–522.

[7] M. Weske, G. Decker, M. Dumas, M. La Rosa, J. Mendling, H.A. Reijers, Model Collection of the Business Process Management Academic Initiative. URL, https://zenodo.org/record/3758705.

[8] J.A.H. López, J.S. Cuadrado, MAR: a structure-based search engine for models, in: Int. Conf. on Model Driven Engineering Languages and Systems, 2020, pp. 57–67.

[9] J.A.H. López, J.S. Cuadrado, An efficient and scalable search engine for models, Softw. Syst. Model. 21 (5) (2022) 1715–1737.

[10] P.T. Nguyen, J. Di Rocco, L. Iovino, D. Di Ruscio, A. Pierantonio, Evaluation of a machine learning classifier for metamodels, Softw. Syst. Model. 20 (6) (2021) 1797–1821.

[11] J.A.H. López, R. Rubei, J.S. Cuadrado, D. Di Ruscio, Machine learning methods for model classification: a comparative study, in: Int. Conf. on Model Driven Engineering Languages and Systems, 2022, in press.