Interoperable Machine Learning Metadata using MEX

Diego Esteves[1], Diego Moussallem[1], Ciro Baron Neto[1], Jens Lehmann[1], Maria Claudia Cavalcanti[2], and Julio Cesar Duarte[2]

[1] University of Leipzig, AKSW , Germany
email: {lastname}@informatik.uni-leipzig.de
[2] Military Institute of Engineering, Department of Computer Engineering, Brazil
email: {yoko,duarte}@ime.eb.br

Abstract. One key step towards machine learning scenarios is the reproducibility of an experiment as well as the interchanging of machine learning metadata. A notorious existing problem on different machine learning architectures is either the interchangeability of measures generated by executions of an algorithm and general provenance information for the experiment configuration. This demand tends to bring forth a cumbersome task of redefining schemas in order to facilitate the exchanging of information over different system implementations. This scenario is due to the missing of a standard specification. In this paper, we address this gap by presenting a built upon on a flexible and lightweight vocabulary dubbed MEX. We benefit from the linked data technologies to provide a public format in order to achieve a higher level of interoperability over different architectures.

1 Introduction

So far, we have seen a variety of publications on the Machine Learning (ML) topics, many of them contributing to the state of the art in their respective fields. However, the last years experienced a *knowledge gap* in the standardization of experiment results for mapping and storing produced performance measures. This technological gap can be summed up by the following question: *"How to achieve interoperability among machine learning experiments over different system architectures?"*. In other words, experimental results are not delivered in a common machine-readable way, causing the information extraction and processing to be tricky and burdensome [1]. Generally, the missing of a consensus for a lightweight and flexible format to achieve the interoperability for machine learning experiments over any system implementation sakes on the development of schema based on existing machine readable formats, using established formats (e.g.: *Extensible Markup Language* (XML), *Comma-separated values* (CSV)), which do not allow high levels of interoperability though. In this paper, we introduce an application program interface (*API*) based on MEX Vocabulary [2] to tackle with this *gap*, allowing the generation of common outputs to be either reused or processed by other systems regardless software implementation and platform (Figure 1). To the best of our knowledge, this is the first report in the literature of an API for exporting metadata of machine learning iterations based on an interchange format.

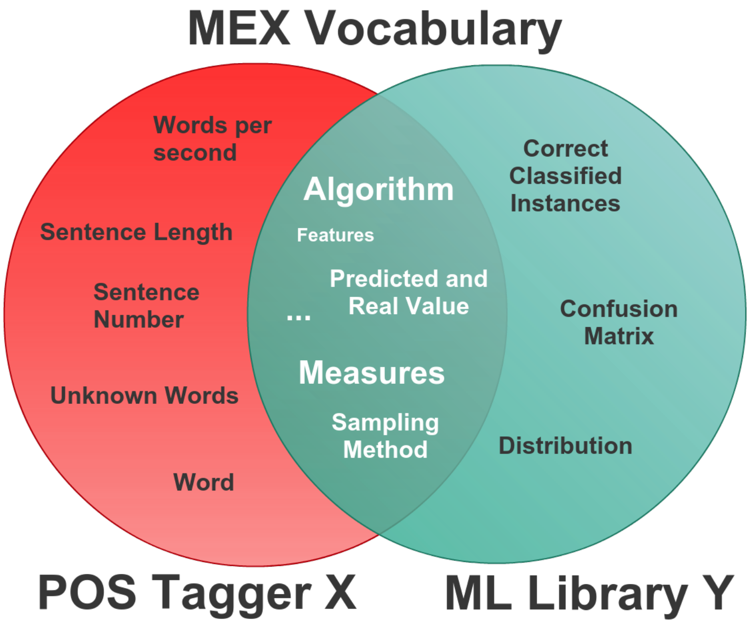


Fig. 1: An interchange format for common variables related with a run of a machine learning algorithm. The intersection of two different machine learning scenarios (part-of-speech and classification problem) using different implementations (X and Y) highlighting examples of frequent machine learning variables between them.

2 MEX Format: a lightweight interchange format

The MEX vocabulary [2] has been designed to tackle the problem of sharing provenance information particularly on the machine learning iterations (for Classification, Regression and Clustering problems) in a lightweight and flexible format, built upon on the *W3C PROV Ontology (PROV-O)* [3], i.e., the format aims to allow the interchange of variables existing on each run of a machine learning algorithm among different systems implementations. The MEX vocabulary is composed by three layers: `mexcore` (*formalizes the key entities for representing the iterations on the machine learning executions, where each iteration has parameters as input and measures as output*) ;`mexalgo` (*represents the context of machine learning algorithms)* ;`mexperf` (*provides the basic entities for representing the associated measures*). Variables concerning the ML pipeline, which often involves a sequence of data pre-processing, model fitting, feature extraction analysis, and validation stages are out of scope for this work. They can be managed properly by implementing an existing scientific workflow system ([4], [5]), however presenting a low level of interoperability for different system architectures. The MEX focuses on a lightweight format of the basic elements for each iteration of a machine learning algorithm in order to achieve a higher level of interoperability: *the performed execution itself and its parameters, as well as the produced measures*.

3 Demonstration

In this demo paper, we show the MEX usage for two different programming-languages: Java[3] and NodeJS[4]. We argue that a higher level of interoperability can be achieved exporting the variables using MEX as a format. The Figure 2 depicts an overview of the system architecture, where the three layers provide the full MEX schema, whereas the Jena API[5] (representing the RDF Library for the Java scenario) represents the RDF serialization. We present the development of the Java and NodeJS APIs [2] and similar use cases as examples, showing the advantage of defining MEX as a format for the machine learning iterations.

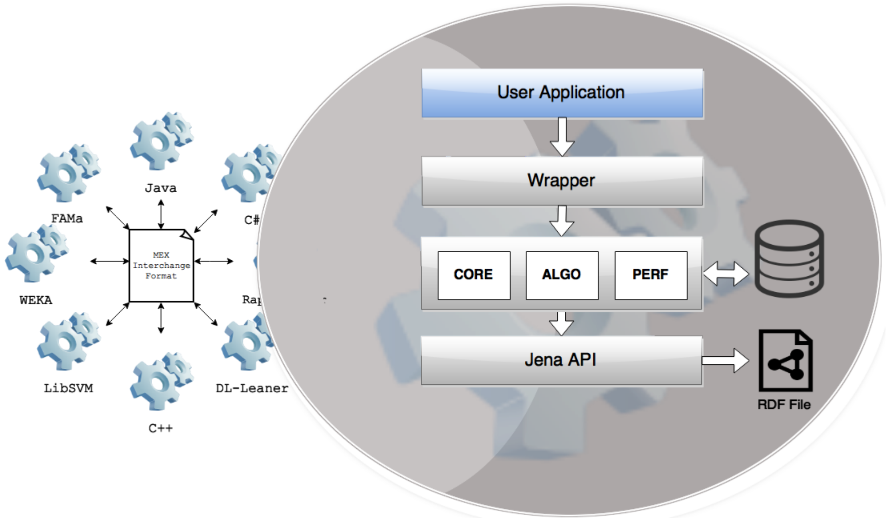


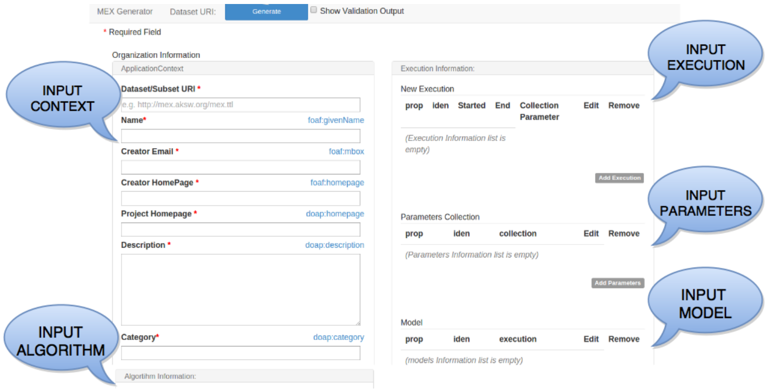
Fig. 2: Achieving a higher level of interoperability with the MEX APIs. A conceptual representation of machine learning open source tools (*Weka, LibSVM, DL-Learner, FAMa, RapidMiner*) and programming language libraries (*Java, C++, C#*), stated as "User Application" exchanging provenance information of machine learning executions through the MEX wrappers (based on the 3 layers of the MEX Vocabulary)

The MEX files for these examples can be found here [2] as well as the use case implementation for Java and Weka. Moreover, to assist with the tedious task of generating *Latex* tables based on the machine learning performance outputs (a manual task commonly executed by the user), we implemented functions to automatize this task based on MEX files[6]. Finally, we also provide a GUI to generate the basic MEX file (Figure 3) for non-expert users[6].

[3] `https://java.com/pt_BR/download/`

[4] `https://nodejs.org/`

[5] `https://jena.apache.org/`



* Required Field

Organization Information

ApplicationContext

Dataset/Subset URI *
e.g. http://mex.aksw.org/mex.ttl

Name* foaf:givenName

Creator Email * foaf:mbox

Creator HomePage * foaf:homepage

Project Homepage * doap:homepage

Description * doap:description

Category* doap:category

Algortihm Information:

Execution Information:

New Execution

prop	iden	Started	End	Collection Parameter	Edit	Remove
(Execution Information list is empty)						

Parameters Collection

prop	iden	collection	Edit	Remove
(Parameters Information list is empty)				

Model

prop	iden	execution	Edit	Remove
(models Information list is empty)				

Fig. 3: The MEX GUI generator for non-expert users

4 Conclusions and Future Work

We defined a novel interface for the representation of the variables associated with the machine learning model executions and developed a Java and NodeJS *APIs* based on that, allowing the exporting of a flexible and lightweight format for data interchanging. As future work, we plan the integration with more established platforms (e.g.: [5]) and new programming languages, such as *Weka*[6] and *C++*[7], for instance. Also, a repository for MEX files linked with nanopublications[8] and the examination of more machine learning representations are desired. Finally, we argue that experiment databases [1] can benefit from the defined MEX interchange format and its APIs.

Acknowledgments This research has been partially supported by grants from the CAPES foundation, Ministry of Education of Brazil, Brasilia - DF 70040-020, Brazil (Bolsista da CAPES - Proc. n: BEX 10179/13-5) and the H2020 ALIGNED Project (GA No. 644055)

References

1. Joaquin Vanschoren et al. Experiment databases. *Machine Learning*, pages 127–158, 2012.
2. Diego Esteves et al. MEX Vocabulary: A lightweight interchange format for machine learning experiments. In *SEMANTiCS 2015*, 2015.
3. W3C PROV-O ontology. `http://www.w3.org/TR/prov-o/`.
4. Yolanda Gil et al. Wings: Intelligent workflow-based design of computational experiments. *IEEE Intelligent Systems*, pages 62–72, 2011.
5. Joaquin Vanschoren et al. Openml: Networked science in machine learning. *SIGKDD Explor. Newsl.*, pages 49–60, 2014.
6. MEX website. `http://mex.aksw.org`.

[6] http://www.cs.waikato.ac.nz/ml/weka/

[7] http://en.cppreference.com/w/

[8] http://nanopub.org/wordpress/