LOG4MEX: Lightweight Machine Learning Metadata Library

Diego Esteves[1], Diego Moussallem[1], Jens Lehmann[2,3], Muhammad Saleem[1], Pablo N. Mendes[5], Patrick Westphal[1], Ciro Baron Neto[1], Julio Cesar Duarte[4], Maria Claudia Cavalcanti[4], Tommaso Soru[1], and Axel-Cyrille Ngonga Ngomo[1]

[1] University of Leipzig, AKSW, Germany
email: {lastname}@informatik.uni-leipzig.de
[2] University of Bonn, Germany jens.lehmann@cs.uni-bonn.de
[3] Fraunhofer IAIS, Germany jens.lehmann@iais.fraunhofer.de
[4] IBM Research Almaden, USA
email: pnmendes@us.ibm.com
[5] Military Institute of Engineering (IME), Brazil
email: {duarte,yoko}@ime.eb.br

Abstract. With the rise of big data, the choice of the best computational solution for a particular task is increasingly reliant on experimentation. Even though experiments are often described in research publications through text, tables and figures, descriptions are often incomplete or very time consuming for researchers to interpret. Researchers often have to perform lengthy web searches to obtain a deep understanding of the software with its configurations and parameters in order to reproduce the results mentioned in publications. In addition, understanding the available data and metadata can be cumbersome, since there are dozens of ways to represent it. In this paper, we present an open-source library dubbed LOG4MEX, which aims to support the scientific community to export machine learning outputs directly from Java code in the MEX interchange format, regardless of the used ML API or the context of experiment. We performed a full analysis of the library in a real scenario of named entity recognition (NER) as well as introduced further use cases, showing the advantages of the proposed resource and the benefits of a semantic format.

Keywords: LOG4MEX, Machine Learning, Interoperability, Provenance, Data Management, Metadata, Reproducible Research

1 Introduction

Nowadays, machine learning (ML) solutions have gained substantial attention due to a number of attractive solutions for existing scientific problems. However, researchers often take longer than expected to get a good understanding and process of all the information available in the solutions, since there are no proper standards to export and share experiment results. According to Peng, "*replication is the ultimate standard by which scientific claims are judged*" [1]. Besides research issues, a major consequence is the negative impact that a product created based on overestimated results of a given research activity can bring to society. A recent worrisome study showed that only 39

out of 100 replication attempts were successful. Furthermore, just 1 out of 61 experiments that could not be reproduced presented a "virtually identical" result [2] during the *Reliability test* [3]. Further analysis found that only 6 of 53 high-profile papers in cancer biology could be reproduced [4]. One of the most famous examples of misleading research we have to date is the *Potti scandal* at Duke University. In this scandal concerning chemotherapy treatment, about 2.000 hours were needed just to find out the mistakes in the experimental setup in addition to the negative impact on society [5].

In accordance with [1], "publication", "code" and "data" should be available in repositories in order to make a research reproducible (RR). Reproducibility allows for people to focus on the actual content of a data analysis, rather than on superficial details reported in a written summary. In terms of his "reproducibility spectrum", we aim to provide means to improve the quality of available data pertaining to machine learning outputs by improving their interoperability, provenance as well as providing a more complete description of the variables and their relationships. Note that in the field of machine learning, results are particularly hard to understand and/or replicate because of commonly missing technical details pertaining to the execution context and parameters of algorithms.

To address this problem, we designed a library called LOG4MEX[6] based on the MEX Vocabulary [6][7]. MEX is an initiative to define standards for the publication of ML experiments in the Semantic Web [8] aiming to reduce the probability of misinterpretation, to facilitate the data management process of the machine learning outputs and to avoid the lack of a more refined schemata definition (unfortunately still a recurring practice).

The remaining paper is structured as follows: Section 2 exemplifies the problem with relation to RR issues. Section 3 progresses to discuss related works. Section 4 details the resources presented in this work. Section 5 shows how the resource can be used in a real world examples. Section 6 points out conclusions and possibilities for future work.

2 Problem

In this section, we show some issues observed in the NER use case in order to exemplify the problems tackled by our framework. These issues are addressed in Section 5.1 using a more refined schemata provided by LOG4MEX. The format of the original data[9] highlights the disadvantages of not having a standard structure defined. Albeit, some assumptions can be made due to the good levels of organisation in the physical folder structure, such as "*that might be the data for the first experiment*" and "*it should be the result for the token-based approach*", these assumptions are still suppositions. Therefore, we list the most relevant issues related to the models and parameters: (i) originally "*Support Vector Machines (SVM)*" were invented by Vapnik in 1963 [7]

[6] https://github.com/AKSW/mexproject/tree/master/log4mex
[7] https://w3id.org/mex
[8] https://www.w3.org/community/ml-schema/
[9] https://github.com/AKSW/FOX/tree/master/evaluation

and nowadays there are many different implementations and *kernel* derivations. Therefore, the acronym *SVM* does not specify *a priori* the correct implementation. (ii) the authors cite the *Weka* framework [8] in order to define the values for the defaults algorithm's parameters. This dependency can lead to misinterpretations once the values for the default parameters can either be changed or not declared by the tool's documentation. Also, the *software version* is missing, making the replication of this task more difficult. (iii) foreseeing the Stanford NER *classifier* is also not suitable pertaining RR recommendations. Depending on the model (`eng.all.3class.distsim`, `eng.muc.7class.distsim` or `eng.conll.4class.distsim`), different results would be achieved (for instance "*A Hologram for the King - Tom Tykwer's latest movie hits German cinemas*"), even though just 3 classes are being used. (iv) some datasets, such as "news" dataset, do not present a valid "landing page", directly connecting to the resource. Guessing here can also be tricky, leading the reader to different datasets containing similar names, such as "Yahoo News Dataset"[10].

3 Related Work

Web Repositories Recently, some web repositories have been released to share general experiment configurations. *RunMyCode* [9] is a platform which enables scientists to openly share the code and data grounded in their research publications. Analogous, *CodaLab* [10] is an open-source platform which has been designed to address reproducible research issues providing an ecosystem for conducting computational research. In addition, *myExperiment* [11] is a repository and social network for the sharing of bioinformatics workflows. Finally, *OpenML* [12] is a repository to upload machine learning experiments. However, it does not provide semantic web capabilities and the flexibility of *SPARQL* queries and it is limited to *Weka *.arff* files as input data format[11].

Libraries LOG4MEX is the first library available to export machine learning outputs directly from Java code in a generic manner, adding provenance information as well as increasing the interoperability level for a given experiment. Within this context, experiments performed over machine learning *APIs* such as *Weka*, *RapidMinner*, *Java-ML*, *RankLib*, *KNIME*, *JSAT* or *Spark* can share the same schemata. Therewith, we can shorten the time required for understanding results computed with these libraries as well as reducing the probability of misinterpretation.

4 Resource Description

The LOG4MEX library is based on Java best practices, producing an enriched metadata file to share configurations of machine learning executions. It stands as a flexible and lightweight library to represent executions of algorithms and the related vari-

[10] `https://webscope.sandbox.yahoo.com/catalog.php?datatype=r&did=75`

[11] `http://www.openml.org/new/data`

Package	Description
org.aksw.mex.log4mex.algo	Direct mappings to mex-algo vocabulary
org.aksw.mex.log4mex.core	Direct mappings to mex-core vocabulary
org.aksw.mex.log4mex.perf	Direct mappings to mex-perf vocabulary
org.aksw.mex.log4mex.perf.example	Classes to represent performance of executions at *example* level `[mexcore:SingleExecution]`
org.aksw.mex.log4mex.perf.overall	Classes to represent performance of executions at *subset* level `[mexcore:OverallExecution]`
org.aksw.mex.util	Static variables to map the vocabulary and control variables
org.aksw.mex.util.ontology	Representation of diverse useful existing ontologies
org.aksw.mex.util.ontology.mex	Basic MEX classes types

Table 1: LOG4MEX Architecture Components:

ables based on the MEX Vocabulary [6][12]. This feature covers an important existing gap in standardization of machine learning approaches. Furthermore, LOG4MEX helps to bridge the gap between the areas of machine learning and semantic web. Diverse areas which implement the flow $input(parameters) \Rightarrow algorithm(models) \Rightarrow output(measure)$ can benefit from the proposed library, such as *natural language processing* and *stock market predictions*. A short description of the architecture components depicted by the Figure 1 is available in Table 1. LOG4MEX is designed to encapsulate semantic web features (Figure 1), making the generation process transparent to the end user. For instance, the ontology package `org.aksw.mex.util.ontology` provides the mappings to upper-level ontologies. Finally, FAIR principles[13] are followed, collaborating along with RRissues.

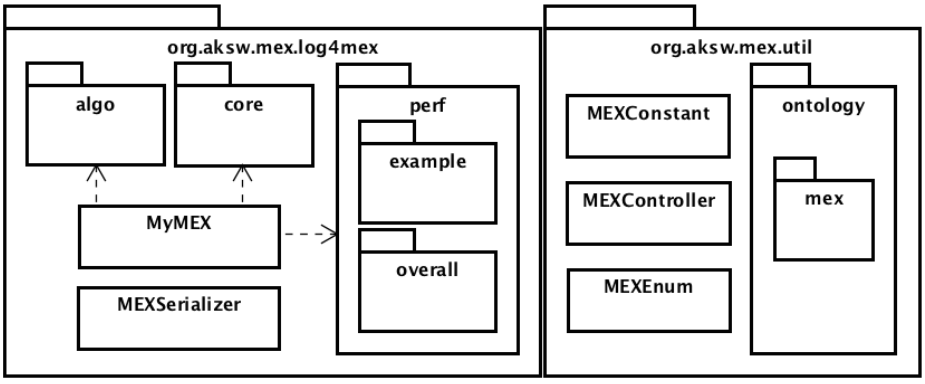

Fig. 1: LOG4MEX component diagram: the modularization designed to keep the flexible and lightweight characteristic of MEX.

[12] `https://w3id.org/mex`
[13] `https://www.force11.org/group/fairgroup/fairprinciples`

Our library has two main classes: `MyMEX` and `MEXSerializer`. The first acts as a complex object to hold the ML variables as well as to add some authoring provenance information pertaining to the experimentation design. Whereas `MEXSerializer` is a *singleton* object responsible to parse and serialize the meta-data (Listing 1.1). The complete documentation can be found at GitHub's project website[14].

```
MyMEX mex = new MyMEX();
mex.setAuthor("R. Speck", "speck@informatik.uni-leipzig.de");
mex.setContext(EnumContexts.NER);
mex.setOrganization("Leipzig University");
mex.setExperimentTitle("Ensemble Learning for NER");
mex.setExperimentId('E006');
mex.setExperimentDate(new Date('2015-08-04'));
mex.Configuration().setTool(EnumTools.WEKA, "3.6.6");
mex.Configuration().addAlgorithm(EnumAlgorithm.NaiveBayes);
mex.Configuration().addAlgorithm(EnumAlgorithm.SVM);
mex.Configuration().Algorithm(EnumAlgorithms.SVM)
.addParameter("C", "1.0");
mex.Configuration().Algorithm(EnumAlgorithms.SVM)
.addParameter("E", "0.001");
//...
MEXSerializer.getInstance().saveToDisk(filename,URIbase, mex,
    MEXConstant.EnumRDFFormats.TTL);
  System.out.println("here you go, the mex file has been
      successfully created: share it ;-)");
```

Listing 1.1: A simple example of use for LOG4MEX: an excerpt of code (`MyMEX` object) demonstrating its friendly interface and ease of use. The library integrated in ML scripts can more efficiently generate enriched metadata in *run-time*.

5 Use Case and Applications

Developers usually try out several algorithms, with several parameters, until they find the best model. Managing all these data becomes an issue.[15] We introduce some applications for LOG4MEX (Section 5.2), evidencing the wide range of applicability of the resource. As a payback, it becomes easier to query for experiments later (e.g. to find the best results after a grid search for best parameters for a model). Interoperability and provenance level are also advantages, as previously discussed. Also, in contrast with the original plot (Section 2), we show the benefits of LOG4MEX to NER use case (Section 5.1).

5.1 Named Entity Recognition - FOX

FOX [13] is a highly accurate open-source framework for named entity recognition. FOX achieves a higher F-measure than state-of-the-art named entity recognition frame-

[14] `aksw.github.io/mexproject/`

[15] we do not consider *workflow systems* here, but general *Integrated development environments* (IDE)

works like Stanford by combining the results of several approaches through ensemble learning technique. We evaluate the approach[16], comparing current and produced output metadata. By using LOG4MEX we intend to guide and provide means to create a more robust representation of the existing variables in the experiment. With respect to the introduced drawbacks, *Algorithms* and their *hyperparameters* are set, respectively, by the methods `mex.Configuration().addAlgorithm(EnumAlgorithm x)` and `mex.Configuration().Algorithm(EnumAlgorithm x).addParameter(id,value)`. Analogously, `mex.Configuration().setTool(EnumTools y, version)` generates the link to the used version of the software and `mex.Configuration().setModel(name)` can be used to specify the classifier. Finally, the link to each dataset associated to each execution (or configuration) is set by `mex.Configuration().setDataSet(url, name)`. Instead of trawling through the directory structure (more than 20.000 files in about 5GB of uncompressed data) and make suppositions, *SPARQL* queries give the answers much easier (Listing 1.2). Figure 2 depicts an excerpt of code for the generated metadata file. Researchers are able to intuitively understand and absorb the relation of the variables in the experiment configuration.

```
SELECT DISTINCT ?name ?algn ?perfn ?fMeasure WHERE {
       ?execution prov:used ?alg;
                  prov:id ?name.
?perfn prov:wasInformedBy  ?execution.
?perfn mexperf:fMeasure ?fMeasure.
?alg rdf:type ?algn
 }
ORDER BY DESC (?fMeasure)
```

Listing 1.2: Straightforward and adaptable solutions with SPARQL queries.

5.2 Further Applications

Federated Query Engines A query that collects results from more than one data sources is called *federated query*. The engines that allow executing federated queries are called *federated engines*. In SPARQL endpoint federation engines evaluation, the query runtime is the central performance measure. In addition, the number of data sources selected, the source selection time, the number of intermediate results, the precision and recall are also important measures to pinpoint the limitations of the federated engines.In this use case, we exported the LargeRDFBench[17] results in to MEX machine readable format. The selected federated engines can be directly compared using simple SPARQL

[16] `https://github.com/AKSW/mexproject/tree/master/examples/src/main/java/log4mex/fox`

[17] `https://github.com/AKSW/largerdfbench`

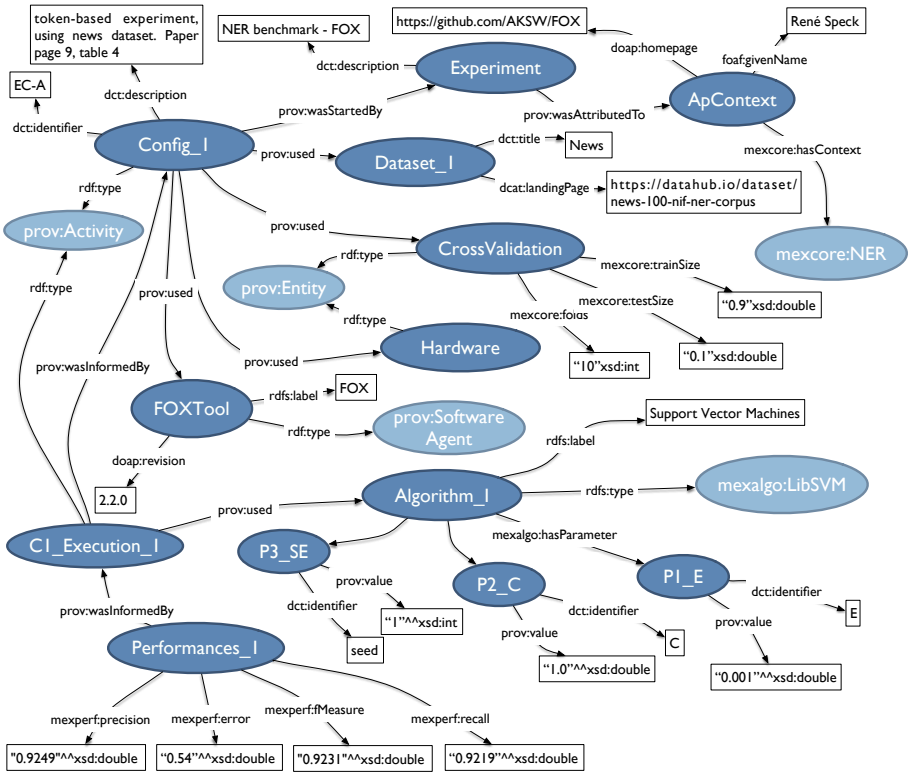

Fig. 2: The visualization of an excerpt of metadata code generated by FOX use case: *dark blue* ellipses representing instance objects and *light blue* for classes. A more refined and semantically enriched metadata structure representing relations in the graph among instances.

queries over MEX results, optimizing the results analysis process. **SML-Bench**[18] proposes a benchmarking framework to run inductive learning tools from the ILP and semantic web communities on a selection of learning problems. Thus, SML-Bench generates results from a systematic selection of benchmarking datasets and learning problems. Here, researchers been working to integrate LOG4MEX into the Framework, reducing the workload to define schemata and manipulate data. **WASOTA**, acronym for *"what are the state-of-the-art for...?"*, is a web repository that encourages researches to keep "state-of-the-art" measures centralized, searching for the best runs based on a domain and specific measure, e.g.: "NER" and "accuracy".

Furthermore, the generated metadata of the above use cases are provided in a SPARQL endpoint[19]. A more complete explanation for each of the introduced example of application for LOG4MEX is available at the resource homepage.

[18] https://github.com/AKSW/SML-Bench
[19] http://mex.aksw.org/sparql

6 Conclusion

In this paper, we contextualize existing problems of managing the generated ML outputs and publishing experiment results, mainly caused due the nonexistent of well defined standards to export and share ML metadata. In order to bridge this gap, we present LOG4MEX, a library that aims to provide an enriched metadata file, facilitating the interoperability of results and data management through semantic web technologies. As use case, we analyzed the impact of the library in the NER context. Results showed the efficiency of the resource, minimizing the overall effort for seeking specific information as well as reducing the probability of misinterpretation. Furthermore, applications which used LOG4MEX were listed, confirming the comprehensiveness of the tool. As future work, we plan to create a REST interface to act as a Middleware to connect the on-the-fly summarized information to Benchmark Systems as well as to integrate the metadata with Web Repositories. Moreover, nanopub servers can be seeing as an interesting resource to connect provenance information about publication and metadata files for machine learning experiments.

References

1. Roger D Peng. Reproducible research in computational science. *Science (New York, Ny)*, 334(6060):1226, 2011.
2. Monya Baker. First results from psychology's largest reproducibility test. nature, 2015.
3. Open Science Collaboration et al. Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716, 2015.
4. C Glenn Begley and Lee M Ellis. Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533, 2012.
5. Darrel Ince. The duke university scandal. *Significance*, 8(3):113–115, 2011.
6. Diego Esteves, Diego Moussallem, Ciro Baron Neto, Tommaso Soru, Ricardo Usbeck, Markus Ackermann, and Jens Lehmann. MEX Vocabulary: A Lightweight Interchange Format for Machine Learning Experiments. In *Proceedings of the 11th International Conference on Semantic Systems*, pages 169–176. ACM, 2015.
7. Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.
8. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
9. Victoria Stodden, Christophe Hurlin, and Christophe Pérignon. Runmycode. org: a novel dissemination and collaboration platform for executing published computational results. In *E-Science (e-Science), 2012 IEEE 8th International Conference on*, pages 1–8. IEEE, 2012.
10. CodaLab. `http://research.microsoft.com/en-us/projects/codalab/`. Accessed: 2016-04-29.
11. Carole A Goble, Jiten Bhagat, Sergejs Aleksejevs, Don Cruickshank, Danius Michaelides, David Newman, Mark Borkum, Sean Bechhofer, Marco Roos, Peter Li, et al. myexperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic acids research*, 38(suppl 2):W677–W682, 2010.
12. Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.
13. René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble learning for named entity recognition. In *The Semantic Web–ISWC 2014*, pages 519–534. Springer, 2014.