OpenAIRE's DOIBoost - Boosting CrossRef for Research

Sandro La Bruzzo[1], Paolo Manghi[1], Andrea Mannocci[2]

[1] Institute of Information Science and Technology - CNR,
Pisa, Italy
{paolo.manghi, sandro.labruzzo}@isti.cnr.it

[2] Knowledge Media Institute - Open University, UK
andrea.mannocci@open.ac.uk

Abstract. Research in information science and scholarly communication strongly relies on the availability of openly accessible datasets of scholarly entities metadata and, where possible, their relative payloads. Since such metadata information is scattered across diverse, freely accessible, online resources (e.g. CrossRef, ORCID), researchers in this domain are doomed to struggle with (meta)data integration problems, in order to produce custom datasets of often undocumented and rather obscure provenance. This practice leads to waste of time, duplication of efforts, and typically infringes open science best practices of transparency and reproducibility of science. In this article, we describe how to generate DOIBoost, a metadata collection that enriches CrossRef with inputs from Microsoft Academic Graph, ORCID, and Unpaywall for the purpose of supporting high-quality and robust research experiments, saving times to researchers and enabling their comparison. To this aim, we describe the dataset value and its schema, analyse its actual content, and share the software Toolkit and experimental workflow required to reproduce it. The DOIBoost dataset and Software Toolkit are made openly available via Zenodo.org. DOIBoost will become an input source to the OpenAIRE information graph.

Keywords: scholarly communication, open science, data science, data integration, CrossRef, ORCID, Unpaywall, Microsoft Academic Graph.

1 Introduction

Research in information science and scholarly communication strongly relies on the availability of openly accessible datasets of metadata and, where possible, of relative payloads. In the context of literature publishing, CrossRef is certainly playing a central role as mediator between publishers of scientific literature and consumers, which are often also producers in this process. Publisher services publish scientific literature, mint a DOI from CrossRef, and push into the system a complete bibliographic record according to the CrossRef metadata scheme. In turn, CrossRef provides CC-BY 4.0 access to its entire metadata collection via REST APIs[1]. Due to its longitudinal, pan-publisher and up-to-date content, this metadata collection has become the pivot of several other initiatives willing to *(i)* enrich/complete the collection with further information, not necessarily provided by publishers to CrossRef, or *(ii)* willing to enrich their own collection with DOIs and metadata from CrossRef. Several well known examples can be mentioned, such as Google Scholar, Dimensions, SemanticScholar, Microsoft Academic Graph, AMiner, OpenAIRE, ORCID, Unpaywall; many of them make their content freely available for research purposes, under CC-BY or CC-0 license. Researchers can download or access via APIs such metadata collections and perform their experiments, but only after non-trivial efforts of (meta)data integration, cleaning, and harmonization. Efforts often given for granted by major players in scholarly knowledge analytics and dismissed in one sentence where a list of data sources, often paywalled and thus not available to the general public, is provided [5]. Typically, such integration efforts differ from experiment to experiment, where, violating principles of Open Science, provenance and lineage of the data are often not documented. This general misalignment spoils quality, and evaluation and comparison of different research endeavours, which should be rather based on common input data collections, transparently generated and recognized by the community.

In response to this general demand, this data paper presents DOIBoost [6], a collection of metadata records resulting from a transparent process of integration, harmonization, and cleaning of *CrossRef* with *Microsoft Academic Graph*[2] (via *Azure Data Lake Store*), *ORCID*,[3] and *Unpaywall*.[4] Such sources can considerably impact on the quality and richness of CrossRef by adding publication access rights information, missing abstracts, author identifiers, and precious authors' affiliations equipped with organization identifiers. The result of our integration efforts, the DOIBoost dataset, is here described, i.e. its input sources, its data model (JSON schema), together with the methodology to generate the dataset, and the actual software (DOIBoost Software Toolkit) and machinery used to produce it. Both DOIBoost dataset and software are published in Zenodo.org [6][7] and made available for research purposes under CC-BY 4.0. DOIBoost will become an input source to the OpenAIRE information graph.[5]

[1] *CrossRef APIs*, https://www.crossref.org/services/metadata-delivery/rest-api/

[2] *Microsoft Academic Graph*, https://aka.ms/msracad

[3] *ORCID,* http://orcid.org

[4] *Unpaywall*, http://unpaywall.org

[5] OpenAIRE EXPLORE, http://explore.openaire.eu

2 The dataset

DOIBoost is constructed by enriching CrossRef records as shown in Figure 1: the input sources described in Table 1 are collected and integrated by using CrossRef DOIs as pivot for the data integration process. A final cleaning step is applied, to get rid of the records whose quality is too low or that are leftovers inserted in CrossRef for testing reasons and never removed. In the following sections, we provide details on the input sources and describe the DOIBoost data model.

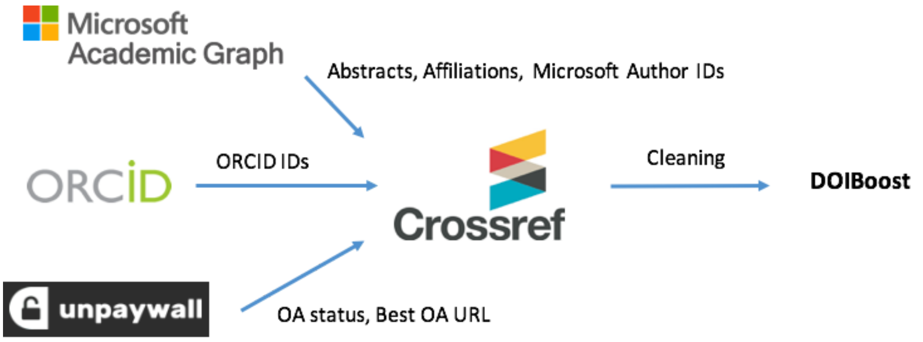

Figure 1 - DOIBoost: dataset construction workflow.

2.1 Input dataset sources

The input data sources are described in Table 1. Their metadata is provided under free-to-reuse and distribute license, although with slightly different constraints, which however do not prevent the dissemination of the collection. Such sources are relevant to CrossRef due to the following reasons:

- *Unpaywall* by ImpactStory [1] attempts to identify the Open Access records in CrossRef by also crawling from the Web (e.g. from institutional repositories) the best Open Access URLs they can find for each record. CrossRef DOIs can be enriched with such Open Access instances.
- *Microsoft Academic Graph, via Azure Data Lake Store (ADLS)* [2] uses "...AI-powered machine readers to process all documents discovered by Bing crawler and extract scholarly entities and their relationships to form a knowledge base...". When possible, MAG links to DOIs and can therefore enrich CrossRef with extra information, e.g. author identifiers, affiliation identifiers, abstracts.

- *ORCID* [3] builds a world-wide record of researchers by providing them with a persistent identifier and allowing them to populate a publicly accessible curriculum, inclusive of article DOIs. As a result, ORCID gathers many more associations between articles in CrossRef and ORCID IDs than CrossRef is actually collecting from publishers.

Table 1 - DOIBoost: input datasets.

Source	**License**	**Protocol & format**	**Approximate size**	**Download date**
CrossRef	CC-BY 4.0	API, JSON[6]	250GB	May 2018
ORCID	CC-0 1.0	Download, CSV (txt)[7]	32GB (zipped)	Dec 2017
MAG (ADLS)	ODC-BY	Download, CSV (txt)[8]	120GB (relevant DB tables)	May 2018
Unpaywall	CC-BY 4.0	Download, CSV (txt)[9]	6GB (zipped)	Dec 2017

2.2 Dataset model

CrossRef, as well as the other sources, are integrated into a common (meta)data model and JSON schema, initially populated with CrossRef records. The model is illustrated via an example in Listing 1.

```
{  "title":"My Title",
   "authors":[
      {  "given":"Marco",
         "family":"Rossi",
         "fullname": "Marco Rossi",
         "identifiers":[
            {  "schema":"ORCID",
               "value":"https://..../0000-0002-3337-2025",
               "provenance":"ORCID" },
            {  "schema":"MAG ID",
               "value":"https://.../1278293695",
```

[6] *CrossRef APIs*, http://api.crossref.org

[7] *ORCID download*, https://orcid.org/content/download-file

[8] *Microsoft Academic Graph* obtained via the *Azure Data Lake Store (ADLS)*, https://azure.microsoft.com/en-us/services/storage/data-lake-storage

[9] *Unpaywall download*, https://unpaywall.org/products/snapshot

```
               "provenance":"MAG" } ],
         "affiliations":[
            {  "value":"My Affiliation Name",
               "official-page":"www.affiliation.org",
               "identifiers":[
                  {  "schema":"grid.ac",
                     "value":"https://.../grid.12345.a" },
                  {  "schema":"microsoftID",
                     "value":"https://.../4213412341" },
                  {  "schema":"wikipedia",
                     "value":"https:///wiki/my_affiliation" }],
               "provenance":"MAG" } ] },
      {  "given":"Giuseppe",
         "family":"Trovato",
         "fullname": ″Giuseppe Trovato",
         "identifiers":[],
         "affiliations":[] } ],
   "issued":"2016-07-01",
   "abstract":[
      {  "value":"Abstract Text", "provenance":"MAG" },
      {  "value":"Abstract Text", "provenance":"CrossRef" } ],
   "subject":["Agronomy and Crop Science", "Forestry"],
   "type":"journal-article",[10]
   "license":[
      {  "url":"http://www.elsevier.com/tdm/userlicense/1.0/",
         "date-time":"2011-07-01T00:00:00Z",
         "content-version":"tdm",
         "delay-in-days":0 } ],
   "instances":[
      {  "url":"http://unkonwonInstance.org",
         "access-rights":"UNKNOWN", "provenance":"CrossRef" },
      {  "url":"http://openAccessInstance.org",
         "access-rights":"OPEN", "provenance":"Unpaywall" } ],
   "published-online":"2016-08-01",
   "published-print":"2016-07-01",
   "accepted":"2016-01-01",
   "publisher":"Publisher Name",
   "doi":"10.1016/j.ffhfhgfhf",
   "doi-url":"http://dx.doi.org/10.1016/j.ffhfhgfhf",
   "issn":[ { "type":"print", "value":"01234-5678" } ],
   "collected-from":[ ‶CrossRef″, "MAG", "Unpaywall", "ORCID" ]
}
```

Listing 1 - DOIBoost: JSON record example.

Data integration is relevance-driven in the sense that from the input data sources only a few properties, regarded as particularly important, are selected for integration into the CrossRef dataset. Accordingly, the model has been conceived to include a set of CrossRef properties, as provided by the relative dataset, and a set of properties that can be integrated from other sources, as described in Figure 1. Each of these “inheritable” properties is equipped with a *provenance* field, whose value currently

[10] *CrossRef record type*, https://api.crossref.org/v1/types

includes “CrossRef”, “MAG”, “Unpaywall”, and “ORCID” in order to trace the origin of the information. Provenance plays a key role when processing a dataset in order to account for the origin of any possible misbehaviour or unexpected result and to fine-tune processing based on the data model and on specific provenance of given fields. Table 2 describes the current property-data source setting, together with a measure of the “boost” that each data source gives to CrossRef DOIs. More specifically, the properties are:

- *Identifiers of authors*: authors can be assigned identifiers from different “authorities”, for example internal identifiers as provided by Microsoft or persistent identifiers as provided by ORCID. Accordingly, the model allows to gather multiple identifiers for the same author; to facilitate programmatic interpretation, for each identifier *value* the model includes the respective *schema*, intended as the authority issuing the identifier.
- *Affiliations of authors*: authors are affiliated to an institution (or more), which is the organization of the author at the moment of publishing; the model allows the collection of different affiliations for the same author since these may be collected from different sources (e.g. CrossRef and MAG); in turn, each institution may be associated to different identifiers provided by the same source, e.g. MAG may provide organization IDs internal to MAG as well as organization persistent identifiers released by the Global Research Identifier Database[11] (hence, same *provenance*, but different *schema*).
- *Abstracts*: abstracts can be provided by different sources, e.g. CrossRef or by MAG, hence require provenance information to track down their origin.
- *Instances of the DOI work*: instances represent the location of the files of a given DOI work at different source sites. Since these may represent different manifestations - e.g. the published journal version, the open access version of an article in an institutional repository - each instance has its own list of files (*URLs*) and *access rights*[12].

Table 2 - Input datasets and properties.

Source	**Properties**	**# of CrossRef DOIs touched by enrichment step**	**Boost: # of enriched CrossRef DOIs**
ORCID	author IDs (ORCID)	7,340,990	6, 052, 611
MAG	DOIs	74,582,104	67,879,513
	affiliation	49,669,576	45,235,583

[11] GRID database, https://www.grid.ac

[12] The field “access-rights” can assume the values OPEN, EMBARGO, RESTRICTED, CLOSED, UNKNOWN.

	(GRID.AC)		
	affiliation (Microsoft)	51,630,810	47,049,156
	abstract	45,407,968	43, 588, 445
	author ID (Microsoft)	74,582,104	66, 036, 936
Unpaywall	instances	90,254,461	11,535,440
CrossRef	all fields	95,567,407	91,365,868 (~4Mi record pruned off because of poor quality)

3 Methodology

Due to the large number of records, in the order of hundreds of millions, our solution relies on in-memory parallel processing techniques. To this aim, the software we developed, named DOIBoost Software Toolkit [7], is deployed over the infrastructure depicted in Figure 2 and reflects the workflows in Figure 3. The architecture workflows support two distinct phases of *(i)* data collection and preparation for integration and *(ii)* data integration to deliver DOIBoost. In the following we described the actions involved in these two phases; knowledge on HDFS and Spark terminology and technologies is strongly advisable to fully understand the internals.

3.1 DOIBoost Toolkit deployment

The infrastructure underlying DOIBoost Toolkit is shown in Figure 3 and features: 20 virtual machines (VMs) for Apache HDFS Data Nodes and Spark workers, each VM with 16 cores, 32GB of ram, and 250GB of disk; plus 3 dedicated virtual machines for HDFS Name Nodes, each one with 8 cores, 16GB of ram, and 40GB of disk.

Apache HDFS is used as the main storage for the objects and files collected from the sources, in order to exploit its fast writing and reading rate. Apache Spark is used to *(i)* read such content from HDFS in order to manipulate and transform it into Spark DataFrames that match the DOIBoost data model, and *(ii)* to perform the data integration pipeline that produces DOIBoost, by joining the data source DataFrames. All workflows are implemented and orchestrated via Apache Oozie[13].

[13] *Apache Oozie*, http://oozie.apache.org

The DOIBoost Toolkit is written in PySpark under AGPL Open Source license[14] and is available for download and citation on Zenodo.org [7]. The package contains the scripts required to implement such workflows and reproduce the collection, which will be described in the following sections.

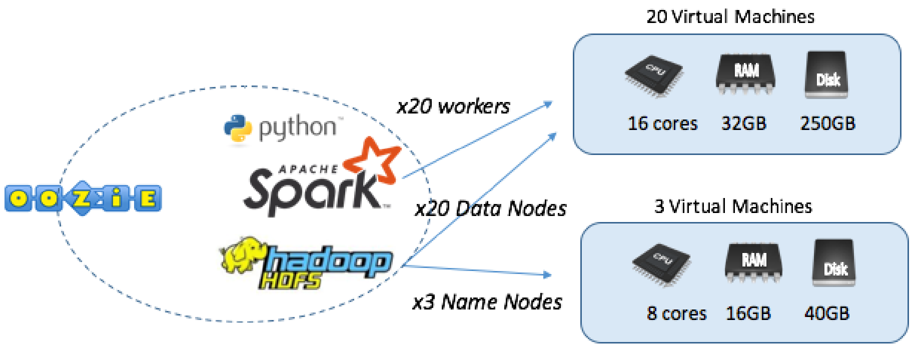

Figure 2 - DOIBoost Toolkit: deployment.

3.2 Data collection and preparation for integration

As anticipated in the previous sections, each source is collected according to different methods, then transferred into HDFS as a corresponding sequence file, and finally manipulated in-memory via Spark jobs (*generateXDataFrame.py*), in order to produce a relative DOIBoost Spark DataFrame.

We assume in the following that the datasets are manually collected and transferred into HDFS via Shell, creating corresponding sequence files: JSON format for CrossRef and CSV text format for MAG, ORCID, and Unpaywall. More specifically:

- CrossRef is downloaded from the relative APIs using the GitHub repository *CrossRef REST API*[15] made available by CrossRef; the execution of the script results in a dump on the file system.
- ORCID and Unpaywall are manually downloaded as CSV text files on the file system. Each line in ORCID represents an author with his/her publication list and in Unpaywall represents a DOI entry with the relative OA status and URL access information.
- MAG is manually downloaded from ADLS as a set of CSV text files, where each CSV is the content of one relational database[16] table in MAG and each line represents a row in the table. For the enrichment of DOIBoost we

[14] *Affero General Public License,* https://en.wikipedia.org/wiki/Affero_General_Public_License

[15] *CrossRef REST API - GitHub*, https://github.com/CrossRef/rest-api-doc

[16] *MAG Schema*, https://microsoftdocs.github.io/MAG/Mag-ADLS-Schema

downloaded content from the following relevant tables: *Papers*, *PapersAuthorAffiliation*, *Authors*, *Affiliation*, *PaperAbstractsInvertedindex.*

Such dumps can be uploaded to HDFS as sequence files with a simple shell command ("hdfs dfs -put fileName pathHDFS"). Once the sequence files are created, the "preparation to integration" phase is performed by executing (in any order) the following Spark jobs:

- *generateCrossRefDataFrame.py* The script reads from the CrossRef sequence file and transforms the JSON records into a respective DOIBoost DataFrame.
- *generateMAGDataFrame.py* The script generates DataFrames corresponding to the MAG tables and performs the joins required to recombine articles with authors, affiliations, and abstracts to deliver a DOIBoost DataFrame that only contains such fields for each article. The process also filters out articles from MAG that do not have a DOI.
- *generateORCIDDataFrame.py* The ORCID sequence file contains rows relative to ORCID author identifiers, each followed by the list of publications of the author. First, the script builds an inverted list, where the key is the DOI followed by the list of authors (the ones that can be found in the sequence file for the DOI) with their first and second names and ORCID ID. Finally, the script builds a DOIBoost DataFrame from these representations, which only contain author information for each article: *given*, *family, fullname, and identifier (schema, value, provenance).*
- *generateUnPayWallDataFrame.py* The Unpaywall sequence file contains rows relative to CrossRef DOIs and their Open Access information[17] as extracted by Unpaywall. The Spark script transforms the file in a DOIBoost DataFrame where articles are equipped with the *instances* derivable from each row (*URL* and *access-rights*) and are empty on other fields.

The execution times of these jobs, given our current architectural specifications, are reported in Table 3 below.

3.3 DOIBoost integration pipeline

Once all DOIBoost DataFrames for the input sources are generated a final integration script can be executed, named *createDOIBoost.py*. The script performs a join by DOI, starting from CrossRef, and adding in sequence: MAG, ORCID, and Unpaywall. Each step of the pipeline progressively enriches DOIBoost DataFrame with one data source at a time (by performing joins on DOIs). Given an input DOIBoost record and one matching its DOI, two kinds of enrichments are possible:

[17] *Unpaywall data format*, https://unpaywall.org/data-format

- *Direct update*: this happens when the joined data source adds incrementally information to the record, such as Unpaywall, which patches the input DataFrame by adding an Unpaywall *instance* to the DOI, and MAG, which adds an *abstract* to the record.
- *Author-match-dependent update*: this happens when the information to be added by the data source is relative to the authors of a DOIBoost record; in this case the authors of the two records to be joined must be matched to find the correspondence and thus complete the information in the proper places; the match is string based and considered positive when the Levenshtein distance[18] between author names is above 0.8.

In both cases, the information is added together with the appropriate provenance information. Finally, the last step of the workflow generates a DOIBoost dump in JSON format on the file system. This step also filters out the DOIBoost records considered of low quality according to the following criteria:

- Absence of key properties: Title or Authors are not provided;
- Basic test records: a record is considered as such if by normalizing the title (i.e. lower-case strings and removal of articles, special characters, etc.) and removing the word "test" the resulting string is empty;
- Structured test records: a record is considered as such if an occurrence of the word "test" appears both in the title and at least in the name of one author.

Table 3 reports on the execution time of the individual steps. Note that the final step *createDOIBoost.py*, performing a join between 95Mi records in CrossRef and 74Mi from MAG, 7 from ORCID, and 90Mi from Unpaywall, runs in around 6 hours.

Table 3 - DOIBoost generation: workflow execution times.

Execution Step	**Execution time**
generateCrossRefDataFrame.py	6.1 m
generateMAGDataFrame.py	1.1 h
generateORCIDDataFrame.py	30 s
generateUnpaywallDataFrame.py	20 s
createDOIBoost.py	6.2 h

[18] *Levenshtein Distance*, https://en.wikipedia.org/wiki/Levenshtein_distance

4 Conclusions

In this paper, we presented our reproducible integration efforts in creating DOIBoost, an open dataset in support of research in the field of scholarly communication and scholarly knowledge mining. The contribution of this work is twofold: first, the dataset itself, together with the description of its value and its content, i.e. the data sources involved in the integration process; secondly, in order not to fall in the perpetration of the infamous "yet another resource" series, the description of the methodology to generate it, embodied in an open source software toolkit that can be used to recreate, extend and update the DOIBoost dataset.

Acknowledgements

This work could be delivered thanks to the Open Science policies enacted by Microsoft, Unpaywall, ORCID, and CrossRef, which are allowing researchers to openly collect their metadata records for the purpose of research under CC0 and CC-BY licenses. The MAG dataset is available with ODC-BY license thanks to the Azure4research sponsorship signed between Microsoft Research and KMi. This work was partially funded by the EU projects OpenAIRE2020 (H2020-EINFRA-2014-1, grant agreement: 643410) and OpenAIRE-Advance H2020 project (grant number: 777541; call: H2020-EINFRA-2017) [4].

References

1. Chawla, Dalmeet Singh. "Unpaywall finds free versions of paywalled papers." Nature News (2017).
2. Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In Proceedings of the 24th International Conference on World Wide Web (WWW'15 Companion). ACM, New York, NY, USA, 243-246.
3. Haak, L. L., Fenner, M. , Paglione, L. , Pentz, E. and Ratner, H. (2012), ORCID: a system to uniquely identify researchers. Learned Publishing, 25: 259-264. doi:10.1087/20120404
4. Manghi, P., Bolikowski, L., Manold, N., Schirrwagen, J., & Smith, T. (2012). OpenAIREplus: the european scholarly communication data infrastructure. D-Lib Magazine, 18(9), 1.
5. Fortunato, S., Bergstrom, C.T., Börner, K., Evans, J.A., Helbing, D., Milojević, S., Petersen, A.M., Radicchi, F., Sinatra, R., Uzzi, B. and Vespignani, A., 2018. Science of science. Science, 359(6379).
6. La Bruzzo, Sandro, Manghi, Paolo, & Mannocci, Andrea. (2018). DOIBoost Dataset Dump (Version 1.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.1438356
7. La Bruzzo, Sandro. (2018, October 1). DOIBoost Software Toolkit (Version 1.0). Zenodo. http://doi.org/10.5281/zenodo.1441058