Version 2
Apr 13, 2021

Investigating DOIs' classes of errors V.2

Ricarda Boente[1], Deniz Tural[1], Cristian Santini[1], Arcangelo Massari[1]
[1]University of Bologna

In Development dx.doi.org/10.17504/protocols.io.bt65nrg6

Open Science 2020/2021

Arcangelo Massari

ABSTRACT

The purpose of this protocol is to describe an automated process to repair invalid DOIs. In particular, four classes of errors are addressed: previously invalid DOIs become valid, prefix errors, suffix errors, and other type errors.

DOI

dx.doi.org/10.17504/protocols.io.bt65nrg6

PROTOCOL CITATION

Ricarda Boente, Deniz Tural, Cristian Santini, Arcangelo Massari 2021. Investigating DOIs' classes of errors. **protocols.io** https://dx.doi.org/10.17504/protocols.io.bt65nrg6 Version created by Arcangelo Massari

CREATED

Apr 13, 2021

LAST MODIFIED

Apr 13, 2021

PROTOCOL INTEGER ID

49085

Checking the DOI names' invalidity

1 We import a CSV containing invalid DOI-to-DOI citations organized as follow: the first column contains valid citing DOIs and the second contains invalid cited DOIs

Citations to invalid DOIs obtained from Crossref

2 First, we check for each cited DOI if it is factually invalid. For this purpose, the DOI Proxy is used (https://www.doi.org/factsheets/DOIProxy.html): if the status code corresponding to that specific DOI is different from 1, it means that it is not valid; otherwise, that DOI has become valid in the meanwhile and the algorithm returns the

Citation: Ricarda Boente, Deniz Tural, Cristian Santini, Arcangelo Massari (04/13/2021). Investigating DOIs' classes of errors. https://dx.doi.org/10.17504/protocols.io.bt65nrg6

same DOI as a correct one.

Error analysis

3 Taking as a reference the DOI error's taxonomy by Buchanan (Buchanan, 2006), there are two main classes of errors: author errors, made by authors when creating the list of cited articles for their publication, and database mapping errors, related to a data-entry error. This protocol deals only with the second kind of error, which can be further divided between prefix errors, suffix errors, and other-type errors (Xu et al., 2019). In order to solve our problem, we isolated recurrent strings at the beginning and at the end with corrupted DOI prefix and suffix respectively. In addition, we found other types of errors, like wrongly encoded characters and unwanted characters, that could be removed at the end of the data cleaning process.

Data cleaning process

4 Here we describe the steps through which we carried out the cleaning process of factually invalid DOIs

4.1 We define two regular expressions: one for identifying corruptions at the beginning of the DOI and one for cleaning corruptions at the end.

4.2 For each factually invalid DOI, we apply the two regular expressions for cleaning prefix-type and suffix-type errors

4.3 Once we have cleaned the DOI from the aforementioned error types we remove unwanted characters, that is double underscores, double periods, XML tags, spaces, and forward slashes.

4.4 After this procedure, we store the modified DOIs in an output CSV file, where for each cited DOI we also store a value corresponding to the error class to which it belongs.

Checking the modified DOI names

5 Finally, each modified DOI name is checked through the DOI Proxy, to verify if the aforementioned procedure was able to fix it or not. In the end, we provide the number of DOI names that we were able to fix and the number of DOI names for each class of errors.

Citation: Ricarda Boente, Deniz Tural, Cristian Santini, Arcangelo Massari (04/13/2021). Investigating DOIs' classes of errors. https://dx.doi.org/10.17504/protocols.io.bt65nrg6