# Cleaning different types of DOI errors found in cited references on Crossref using automated methods

Ricarda Boente, Arcangelo Massari, Cristian Santini, Deniz Tural

Open Science (A.Y. 2020/2021)
Second Cycle Degree in Digital Humanities and Digital Knowledge
Alma Mater Studiorum - Università di Bologna

## Abstract

### Purpose
The purpose of this work is to find an automated process to repair invalid DOI names that have been collected by Silvio Peroni while processing data provided by Crossref (2021).

### Design/methodology/approach
The data needed for this research is provided as a CSV list containing more than 1 million invalid cited DOI names. First, to determine an automated process, the errors that characterize the wrong DOI names in the list need to be classified. Concentrating exclusively on the factual errors, such as additional or invalid characters, the DOI names that have become valid in the meantime can be removed. Then, a classification of those factual errors as prefix-, suffix- or other-type errors is proposed. By closer investigation and extension of already existing research in this field, this research classifies regular expressions that can be used to clean the different types of invalid DOI names: for example, by deleting additional strings at the end or the beginning. After the cleanup, the cleaned DOI names are checked for their validity again.

### Findings
This research was able to find automated processes based on regular expressions and correct the factual errors belonging to different subclasses. Applying the proposed algorithm to the mentioned dataset, around 16% of the DOI names proved valid afterwards. The largest part of those valid DOIs consists of those made valid by cleaning up suffix errors; however, many DOIs also proved valid without cleaning, being only temporarily invalid.

### Research limitations/implications
Checking if the DOI names are valid either consumes a lot of time or a high amount of RAM, since the process should be executed before and after the cleaning. Therefore, the described methods are only applicable on smaller datasets, unless the availability of the necessary resources is ensured. Also, there will always remain DOI names that cannot be made valid using automated processes. In these cases, it is important to find the publishers responsible for the incorrect references, which is done in a separate related project (Cioffi et al., 2021).

### Originality/value
Building on existing research, this study extends and improves regular expressions targeted to clean DOI errors, to enhance the data quality in the COCI dataset. As the COCI project provides open access to reference lists of scientific works, the whole academic community can profit from this improvement in data quality. In addition, the methods submitted could be the base for further research in this field, allowing the correction of DOI name errors in other datasets, too.

## Introduction

Aiming for faster, more transparent, available, and reliable practices in research, the open science movement demands open access not only to scholarly articles but also to their reference lists. With the objective of open scholarly citations, the Initiative for Open Citations (IO4C, https://i4oc.org/) has successfully convinced academic publishers in the last years to make their reference lists available by the means of the platform Crossref (https://www.crossref.org/). Based on the Crossref database, the OpenCitations Index Of Crossref Open DOI-To-DOI Citations, short COCI, was built up, concentrating on DOI-to-DOI citations, and making use of Semantic Web technologies (Heibi et al., 2019). As the data provided by publishers to Crossref is not double-checked, the references listed there can still contain errors, which, if integrated into COCI, would mean a drawback for the project's goal of open scholarly citations. To prevent erroneous data from ending up in the dataset, COCI validates the DOIs using the DOI API and excludes the ones that prove as invalid. A CSV list of these excluded DOIs was provided in the dataset *Citations to invalid DOI-identified entities obtained from processing DOI-to-DOI citations to add in COCI* (Peroni, 2021). This is where this research comes into play. Trying to classify the errors that cause the DOIs to be identified as invalid and finding automatic processes to obtain the correct DOIs could fix millions of citations that would otherwise be lost. Made open to the public, many researchers could profit from the enlarged and more accurate reference lists to build their research on a more extensive base.

The problem of DOI indexing by bibliometric databases is well known in the Scientometrics field. A taxonomy of the main errors was proposed in the article *Accuracy of Cited References: The Role of Citation Databases* (Buchanan, 2006) which distinguished between author errors and database mapping errors. Author errors are defined as inaccuracies caused by authors while creating the list of articles cited for their publications. They can consist of, for example, typos in the name and initials of the first author, in the publication date, in the title, in the volume number or the pagination. On the other hand, database mapping errors are defined as failures in creating an electronic link between a cited article and the corresponding citing article, which can be attributed to a data entry error. These are, for example, transcription errors, omissions of cited articles or in general errors in the registration of the source or target.

| *Error type* | *Author errors* | *Database mapping errors* |
|---|---|---|
| *Definition* | Errors made by authors when creating the list of cited articles for their publication | Failure to establish an electronic link between a cited article and the corresponding citing articles that can be attributed to a data -entry error |
| *Examples* | Errors in name and initials of the first author | Transcription errors |
| | Errors in the publication title | Target-source article record errors |
| | Errors in publication year | The cited article omitted from a cited article- list |
| | Errors in volume number | Reason unknown |
| | Errors in pagination | |

*Table 1 Classification of bibliometric database errors according to Buchanan (2006).*

Other types of errors may be, more in general, classified as human-made errors (e.g. those introduced by editors when managing the references across publications).
However, as the source of DOI errors is not the focus of this work, another approach to define classes of errors was chosen, using the taxonomy proposed in the article *Types of DOI errors of cited references in Web of Science with a cleaning method* (Xu et al., 2019), which distinguishes between errors in the DOI prefix, suffix-errors, and other-type errors. For instance, errors in the prefix are due to the presence of the HTTP or HTTPS

scheme and DOI proxy server domain, that is “dx.doi.org” or “doi.org”. Errors in the suffix, instead, can be the presence of queries to the proxy server, a URL, double DOI names, the indication of the year, hashes, delimiters, and others that will be listed in detail in the *Materials and Methods* section. Finally, double underscores, double periods, XML tags, spaces and forward slashes are considered other-type errors. The article also proposes a regular expression-based methodology to correct these three types of errors, which will be reused and refined in this paper. In addition, the case discussed in *DOI errors and possible solutions for Web of Science* (Zhu et al., 2019) will be addressed, namely that of transcription errors due to the confusion of the letter "O" and the number "0".

The goal of this research is to find answers to the following questions:

- What are the classes of errors that characterized invalid DOIs?
- Which classes of errors can be addressed through automatic processes to get the correct DOIs?
- How many correct DOIs and, consequently, citations have been obtained by applying such automatic processing?

Consequently, the taxonomy of DOI name errors present in the considered dataset will be illustrated first, in the section *Materials and Methods*. Then, still in the same section, the automatic cleaning methods for some of the detected subclasses of errors will be defined and their ability to clean up the DOI names will be measured, presenting the numbers derived in the *Results* section. This work intends to detect some common classes of DOI errors and to provide a way to clean a respectable amount of them by applying automatic methods. Therefore, future Scientometrics-related research could profit from the taxonomy and methods described here. Moreover, even thematically unrelated research could benefit from these findings, given that an improvement in the quality of reference data can be useful to all disciplines. At the end of this article, in the *Discussion and Conclusions* section, an interpretation of the results will be presented, and possible drawbacks will be discussed, as well as an outlook to possible future research.

## Materials and Methods

This research aims to investigate the classes of errors that characterize DOI names of cited publications made available by Crossref. A list of invalid cited DOIs was collected and made publicly available by Silvio Peroni (2021) on Zenodo. The data consists of a CSV file, where each row is a pair of *valid citing DOI – invalid cited DOI*, see sample in Table 2. The file contains a total of 1,223,296 DOI-to-DOI citations. All data used in this research will be handled according to the data management plan provided on Zenodo (Boente et al., 2021a).

| ***Valid citing DOI*** | ***Invalid cited DOI*** |
|---|---|
| '10.14778/1920841.1920954' | '10.5555/646836.708343' |
| '10.5406/ethnomusicology.59.2.0202' | '10.2307/20184517' |
| '10.1177/1179546820918903' | '10.3748/wjg.v10.i5.707.' |

*Table 2 Sample of the CSV file provided by (Peroni, 2021).*

The second input for this research is the *Public Data File from Crossref* (Crossref, 2021), which contains over 120 million metadata records in JSON format. Its usefulness derives from the fact that DOI names at the work level are directly assigned by Crossref and not by publishers, thus always being correct. Therefore, this information can be used to make the DOI validity check procedure dramatically faster in comparison to a validity check using the DOI Proxy Server. However, the higher speed comes at the expense of higher RAM consumption. For this reason, the algorithm, as will be shown in the following, considers as optional the use of this information, and the user is left with the choice of whether to get a result more quickly or whether to sacrifice more RAM.

The goal of this study is to propose an automatic method both for cleaning and classifying the types of errors that occur in the invalid citation data. To do so, this paper proposes:

1. A taxonomy of DOI errors for classifying invalid DOIs.
2. An automatic system designed for obtaining a list of valid DOIs from the CSV file described above and for classifying cleaned DOIs with the respective classes of errors.
3. A visualization procedure for presenting the results.

The basic assumption of this work is that the cited invalid DOIs in the data can be generally ascribed to two categories: DOIs which were temporarily invalid at the time of the data collection and have become valid before the start of our research, and DOIs which are factually invalid (e.g., contain errors) and need to be corrected.

To design this system, this research takes inspiration from Xu et al. (2019) who presented a cleaning method based on regular expressions for removing DOI errors in Web of Science. In this paper, their classification of factual errors in prefix-type, suffix-type and other-type errors is reused and a refined implementation of their cleaning method – enriched for the additional patterns which occur in our data – is presented. Lastly, two steps are added to the pipeline presented by Xu et al. (2019): one at the beginning to check the validity of the cited DOI, and one at the end to check the validity of its cleaned version. Both the validation steps were carried out by using a valid DOI set derived from the *Public Data File from Crossref* and the response provided by the DOI proxy server REST API.

To classify the invalid cited DOIs in the CSV, concerning their classes of errors, it was decided to organize the errors into a taxonomy. As previously stated, this study assumes that there are two main classes of errors: DOI which are temporarily invalid and DOIs which are factually invalid. Additionally, the second class can be split into three further subclasses, as introduced by Xu et al. (2019):

- **DOIs with prefix-type errors**: some DOIs cannot be resolved by the DOI system because they present additional characters *before* the specific sequence "10.XXXX/XXXXXXXXXXX" (e.g. "http://dx.doi.org/10.1016/j.aca.2006.07.086")
- **DOIs with suffix-type errors**: some DOIs cannot be resolved by the DOI system because they present additional characters *after* the specific sequence "10.XXXX/XXXXXXXXXXX" (e.g. "10.1186/1735-2746-10-21,http://www.ijehse.com/content/10/1/21"
- **DOIs with other type errors**: some DOIs cannot be resolved by the DOI system because they contain wrongly added characters, such as double slashes, double underscores, or XML tags.

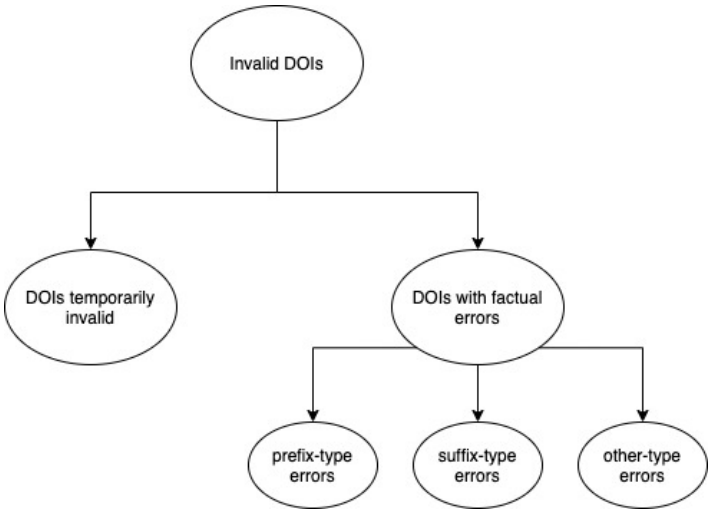


*Figure 1 DOIs errors' taxonomy.*

Given this taxonomy, the objective is to devise the most appropriate algorithm to classify invalid DOIs and to eventually correct them (if needed). To achieve this goal, patterns of factual errors were manually checked in the data considered. A list of recurrent errors in cited DOIs in the dataset created by Peroni (2021) is listed in Table 3, with the regular expressions needed to match the errors:

| ***Invalid DOI*** | ***Correct DOI*** | ***Type of error*** | ***Regular expression*** |
|---|---|---|---|
| 10.1016/J.AMEPRE.2015.07.017. | 10.1016/J.AMEPRE.2015.07.017 | Suffix-type | (.*?)(?:[\.\|,\|<\|&\|\(\|;]+)$ |
| 10.1186/1735-2746-10-21,HTTP://WWW.IJEHSE.COM/CONTENT/10/1/21 | 10.1186/1735-2746-10-21 | Suffix-type | (.*?)(?:[\.\|\(\|,\|;]*HTTP:\/\/.*?)$ |
| 10.1016/J.JLUMIN.2004.10.018.HTTP://DX.DOI.ORG/10.1016/J.JLUMIN.2004.10.018 | 10.1016/J.JLUMIN.2004.10.018<br>10.1016/J.JLUMIN.2004.10.018 | Prefix-type | (.*?)(?:\.)?(?:HTTP:\/\/DX\.D[0\|O]I\.[0\|O]RG\/)(.*))$ |
| 10.1093/BIOINFORMATICS/BTV421.HTTPS://DOI.ORG/10.1101/GR.186072.114 | 10.1093/BIOINFORMATICS/BTV421 | Prefix-type | (.*?)(?:\.)?(?:HTTPS:\/\/D[0\|O]I\.[0\|O]RG\/)(.*) |
| 10.1016/J.TIBS.2006.12.007.....32,63(2006) | 10.1016/J.TIBS.2006.12.007 | Suffix-type | (.*?)(?:\.{5}.*?)$ |
| 10.1021/BI3013565(2012) | 10.1021/BI3013565 | Suffix-type | (.*?)(?: \(\d{4}\)?)$ |
| 10.1287/ORSC.2016.1092>ACCESSED27 | 10.1287/ORSC.2016.1092 | Suffix-type | (.*?)(?:[>\|\)](LAST)?ACCESSED\d+)$ |
| 10.1111/J.1536-7150.2006.00482.X/FULL>ACCESSED4 | 10.1111/J.1536-7150.2006.00482.X | Suffix-type | (.*?)(?:\/(META\|ABSTRACT\|FULL\|EPDF\|PDF\|SUMMARY)([>\|\)](LAST)?ACCESSED\d+)?)$ |
| 10.1007/3-540-35074-8_16#PAGE-1 | 10.1007/3-540-35074-8_16 | Suffix-type | (.*?)(?: #.*?)$ |
| 10.1371/JOURNAL.PONE.0112567.PMID:25405489 | 10.1371/JOURNAL.PONE.0112567 | Suffix-type | (.*?)(?:[\.\|\(\|,\|;]?PMID:\d+.*?)$ |
| 10.1063/1.1148310?CRAWLER=TRUE | 10.1063/1.1148310 | Suffix-type | (.*?)(?: \?.*?=.*?)$ |
| 10.1177/0004865814524218ANJ.SAGEPUB.COM | 10.1177/0004865814524218 | Suffix-type | (.*?)(?:[\.\|\(\|,\|;]?[A-Z]*\.?SAGEPUB.*?)$ |
| 10.1073/PNAS.1104391108[DOI] | 10.1073/PNAS.1104391108 | Suffix-type | (.*?)(?: \[DOI\].*?)$ |
| 10.1073/PNAS.1319051111/-/DCSUPPLEMENTAL | 10.1073/PNAS.1104391108 | Suffix-type | (.*?)(?:\/-\/DCSUPPLEMENTAL)$ |
| 10.1890/15-0075.1/SUPPINFO | 10.1890/15-0075.1 | Suffix-type | (.*?)(?:\/SUPPINF[0\|O](\.)?)$ |
| 10.1101/GR.229202.ARTICLEPUBLISHEDONLINEBEFOREMARCH2002 | 10.1101/GR.229202 | Suffix-type | (.*?)(?:[\.\|\(\|,\|;]?ARTICLEPUBLISHEDONLINE.*?\d{4})$ |

| | | | |
|---|---|---|---|
| 10.1016/J.JPROT.2014.03.043(EPUBAHEADOFPRINT) | 10.1016/J.JPROT.2014.03.043 | Suffix-type | (.*?)(?:[\(|\[]EPUBAHEADOFPRINT[\)\]])$ |
| 10.1186/1471-2407-13-87<br/> | 10.1186/1471-2407-13-87 | Other-type | re.sub("<.*?/>", "", doi) |

*Table 3 List of recurrent factual errors in DOIs (wrong part of DOI highlighted in red), with respective classes of error and regular expressions to clean them.*

Once the recurrent strings that were considered errors in DOI prefixes, suffixes, and of other type were identified, a five-phase automatic data cleaning procedure was designed. A detailed description of this workflow can also be retrieved on protocols.io (Boente et al., 2021b).

- The CSV file containing the invalid DOIs and, optionally, a CSV previously generated from the *Public Data File from Crossref* - containing all the DOIs obtained by Crossref at the work level - are imported.
- The file is checked for cited DOIs which are already valid before the cleaning procedure. Then, they are stored as valid DOI names and labelled as already valid.
- The DOI names which remained invalid after the second step are cleaned using regular expressions.
- The validity of each cleaned DOI is checked through the DOI proxy server or, optionally, a set derived from the *Public Data File from Crossref*. If valid, it is stored as a valid DOI and is labelled with the classes of errors that were cleaned.
- Finally, an enriched version of the input CSV is output. This version contains, for each row, the information related to the validity status of the cited DOI, its valid version (if it was possible to obtain one), and the classes of errors that were eventually cleaned.

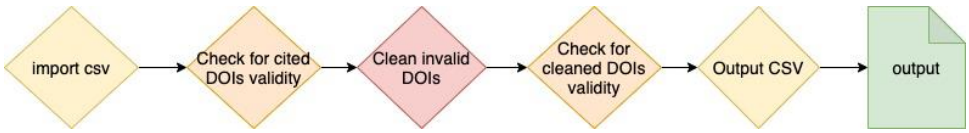


*Figure 2 Visual workflow of the data cleaning procedure.*

Finally, a visualization of the performances of this system is provided, counting the number of valid DOIs which were obtained from this procedure, the number of DOIs which were already valid before cleaning, and the number of errors for each type that occurred in the dataset.

In the following paragraphs, the steps through which we carried out the data cleaning procedure for the invalid cited DOIs are described more closely. The general workflow was presented in the previous section. Python software was created to handle the whole procedure. The source code has been made publicly available with ISC license on Zenodo by Massari et al. (2021).

First, two regular expressions were created: one for cleaning patterns of errors in prefixes of DOIs and another for cleaning suffix-type errors in DOIs. The regular expressions were created by following the example by Xu et al. (2019): the regular expressions which matched the type of errors that we listed in Table 3 were combined as alternatives into a non-capturing group.

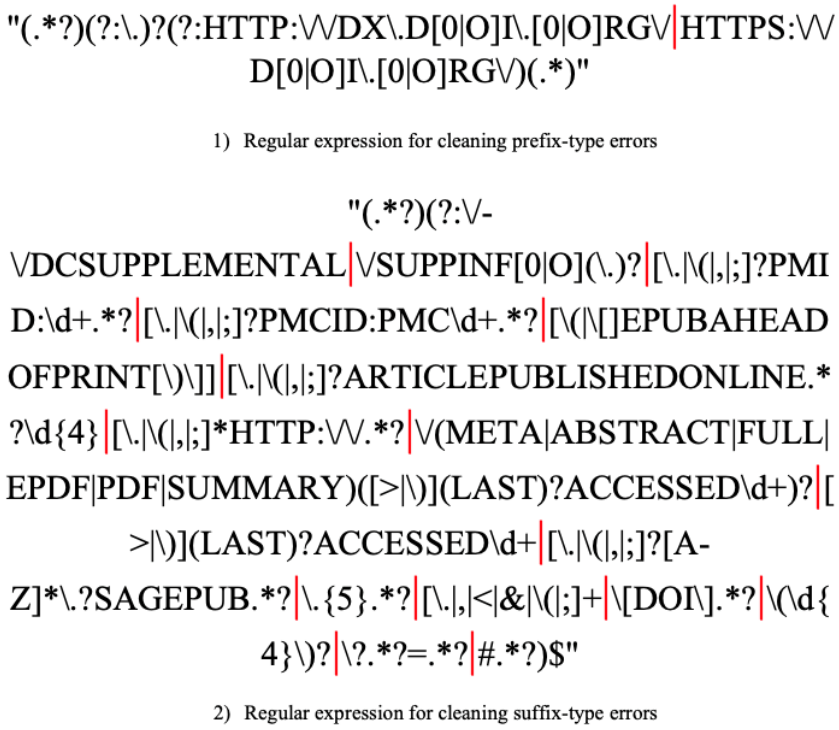


*Figure 3 Regular expressions for cleaning prefix-type and suffix-type errors.*

By comparing these regular expressions with those used in the data cleaning procedure proposed by Xu et al. (2019), it must be mentioned that slight modifications were applied to their structure, to catch additional patterns of errors in the data of this study. First, concerning the regular expression for cleaning prefix-type errors, another capturing group was added at the beginning, on top of the one at the end. This was done because, as shown in row 3 of Table 3, there are cases in which the invalid cited DOI may contain two different DOIs and, in this way, any eventual string which occurs before the prefix error is saved, to be subsequently compared to the string after the prefix error. Furthermore, a pattern was added to capture the new public DOI proxy server, "https://doi.org", in addition to the still supported previous syntax "dx.doi.org", which is no longer preferred. Finally, it is worth noting that in this regular expression, as in the following ones, the character "O" and the number "0" both count as a match, since, as demonstrated in *DOI errors and possible solutions for Web of Science* (Zhu et al., 2019), the two characters are often confused. It is possible to observe the number of occurrences for each of the two prefix patterns in Table 4, noting that the new pattern does not have a very high significance in the dataset:

| ***Regular expression*** | ***# Occurrences*** |
|---|---|
| (.*?)(?:\.)?(?:HTTP:\/\/DX\.D[0\|O]I\.[0\|O]RG\/)(.*) | 21,983 |
| (.*?)(?:\.)?(?:HTTPS:\/\/D[0\|O]I\.[0\|O]RG\/)(.*) | 94 |

*Table 4 Number of matches for the two regular expressions related to prefix errors.*

On the other hand, concerning the regular expression for cleaning errors in suffixes, 9 new patterns were inserted. The regular expressions added to the suffix-type error non-capturing group are highlighted in red in Table 5, with the respective number of matches for the data shown in the right column:

| *Regular expression* | *# Occurrences* |
|---|---|
| (.*?)(?:[\.|,|<|&|(|;]+)$ | 120,828 |
| (.*?)(?: [\.|\(|,|;]*HTTP:\/\/.*? )$ | 22,197 |
| (.*?)(?:\/(META\|ABSTRACT\|FULL\|EPDF\|PDF\|SUMMARY)([>|\)](LAST)?ACCESSED\d+)?)$ | 2,929 |
| (.*?)(?:\(\d{4}\)?)$ | 1,652 |
| (.*?)(?:\.{5}.*?)$ | 565 |
| (.*?)(?:#.*?)$ | 538 |
| (.*?)(?:\?.*?=.*?)$ | 228 |
| (.*?)(?:\[DOI\].*?)$ | 94 |
| (.*?)(?: [\.|\(|,|;]?PMID:\d+.*?)$ | 81 |
| (.*?)(?:[\.|(|,|;]?[A-Z]*\.?SAGEPUB.*)$ | 66 |
| (.*?)(?:[>|)](LAST)?ACCESSED\d+)$ | 65 |
| (.*?)(?:SUPPINF[0|O](\.)? )$ | 49 |
| (.*?)(?:\/-\/DCSUPPLEMENTAL)$ | 13 |
| (.*?)(?:[\.|\(|,|;]?ARTICLEPUBLISHEDONLINE.*?\d{4})$ | 8 |
| (.*?)(?: [\(|\[]EPUBAHEADOFPRINT[\)\]])$ | 5 |
| (.*?)(?: [\.|\(|,|;]?PMCID:PMC\d+.*?)$ | 0 |

*Table 5 Regular expressions for cleaning suffixes, and the relative number of occurrences in the list of invalid cited DOIs. The expressions marked in red have been added in the scope of our research, extending the ones proposed by Xu et al. (2019) (here in black)*

Some minor changes were made to the other regular expressions included in the non-capturing group for suffix errors, such as changes in the occurrence and type of delimiters between the DOI and the erroneous part of the string at the end.

Finally, as regards the other-type errors, substitutions have been made to remove the wrong characters or to replace the repetitions, such as the double underscore, with a single character. Table 6 shows the number of matches obtained for each other-type error sub-category.

| *Regular expression* | *# Occurrences* |
|---|---|
| re.sub("\.\.", ".", doi) | 1,416 |
| re.sub("\\\\", "", doi) | 1,030 |
| re.sub("<.*?/>", "", doi) | 40 |
| re.sub("__", "_", doi) | 38 |
| re.sub("<.*?>.*?</.*?>", "", doi) | 2 |

*Table 6 Number of matches for the regular expressions related to other-type errors.*

Once the regular expressions were devised, the algorithmic solution was implemented by following the steps in Figure 2. More specifically, the implemented solution involves the following phases:

- The CSV file is transformed into the input in a list which maps each row of the file to a dictionary having two keys: the “Valid_citing_DOI” key, whose value is the valid citing DOI in the first column, and “Invalid_cited_DOI” key, whose value is the invalid cited DOI in the second column.
- A function is applied to the list of citing-cited DOIs, to check the validity of the cited ones. If the Crossref DOIs set was created, this is done first by searching if the DOI is present in the set. As mentioned, this procedure was implemented to reduce the computational time required by performing a vast number of HTTP requests to external APIs, since it was assumed that the DOIs registered by Crossref are valid. If the function is not able to find a matching DOI in the set of valid DOIs, it launches a following GET request to the DOI proxy server Rest API, to understand if the DOI could be resolved

by the DOI system (e.g., was not registered by Crossref) or not. As output, the function maps the list of dictionaries in input to an enriched list of dictionaries, with two additional fields: "Valid_DOI", which stores a validated DOI, and "Already_Valid", which stores 1 if it was validated in the meanwhile, 0 otherwise.

- Subsequently, a cleaning function is applied to those cited DOIs which were not labelled as already valid. This function uses the aforementioned regular expressions to remove prefix-type, suffix-type and other types of errors with some additional rules. After the cleaning function is applied, each of the newly obtained DOIs is checked to see if it is different from that in the input data. Then, its validity is inspected with the function used in the previous step. As a result, this function maps the enriched list of dictionaries in input to another list of dictionaries, containing 7 fields: the four already present in the input list ("Valid_citing_DOI", "Invalid_cited_DOI", "Valid_DOI", "Already_Valid") and 3 other fields: "prefix_error", "suffix_error" and "other-type", which contain, for each cleaned DOI, the number of errors that were cleaned.
- After performing the procedure described, the dictionary executed as output in the previous step is stored in an external CSV file, which looks like the one in Figure 4:

| Valid_citing_DOI | Invalid_cited_DOI | Valid_DOI | Already_valid | Prefix_error | Suffix_error | Other-type_error |
|---|---|---|---|---|---|---|
| 10.14778/1920841.1920954 | 10.5555/646836.708343 | | 0 | 0 | 0 | 0 |
| 10.5406/ethnomusicology.59.2.0202 | 10.2307/20184517 | | 0 | 0 | 0 | 0 |
| 10.1161/01.cir.63.6.1391 | 10.1161/circ.37.4.509 | | 0 | 0 | 0 | 0 |
| 10.1177/1179546820918903 | 10.3748/wjg.v10.i5.707. | 10.3748/WJG.V10.I5.707 | 0 | 0 | 1 | 0 |
| 10.1080/10410236.2020.1731937 | 10.1070/10810730903528033 | | 0 | 0 | 0 | 0 |
| 10.1161/strokeaha.112.652065 | 10.1161/str.24.7.8322400 | | 0 | 0 | 0 | 0 |
| 10.1177/1049732310393747 | 10.1111/j.545-5300.2003.42208.x | | 0 | 0 | 0 | 0 |
| 10.1155/2017/1491405 | 10.3760/cma.j.issn.0366-6999.20131202 | | 0 | 0 | 0 | 0 |
| 10.1161/01.res.68.6.1549 | 10.1161/res.35.2.159 | | 0 | 0 | 0 | 0 |
| 10.4018/978-1-5225-2650-6.ch006 | 10.1002/per | | 0 | 0 | 0 | 0 |
| 10.1145/2525314.2594229 | 10.5555/1873601.1873616 | | 0 | 0 | 0 | 0 |
| 10.1007/s10619-020-07320-z | 10.3390/sym11070911www.mdpi.com/journal/symmetry | | 0 | 0 | 0 | 0 |
| 10.1007/s11771-020-4410-2 | 10.13745/j.esf.2016.02.011 | 10.13745/J.ESF.2016.02.011 | 1 | 0 | 0 | 0 |
| 10.1161/01.cir.102.5.591 | 10.1161/circ.85.3.1537115 | | 0 | 0 | 0 | 0 |
| 10.1007/s40617-018-00299-1 | 10.1901/jaba.2012.45-657 | 10.1901/JABA.2012.45-657 | 1 | 0 | 0 | 0 |
| 10.1074/jbc.m508416200 | 10.1059/0003-4819-100-4-483 | | 0 | 0 | 0 | 0 |
| 10.1177/2054358119836124 | 10.1016/j.amepre.2015.07.017. | 10.1016/J.AMEPRE.2015.07.017 | 0 | 0 | 1 | 0 |
| 10.1002/2014jd022782 | 10.1016/j.atmosenv.2008.0305 | | 0 | 0 | 0 | 0 |
| 10.1161/01.cir.97.6.553 | 10.1161/circ.66.1.7083497 | | 0 | 0 | 0 | 0 |
| 10.1080/13642537.2013.849275 | 10.1080/1364253032000157166 | | 0 | 0 | 0 | 0 |

*Figure 4 Sample of the output CSV produced by the cleaning procedure.*

After obtaining the results in this way, an additional manual step was added to measure the correctness of the cleaned DOI names. An appropriate cleaning algorithm should not only make invalid DOI names valid, but these DOIs should also point to the correct reference. For this reason, a random subset of 100 cleaned DOIs was extracted and each DOI was manually checked to verify whether the resolved article was part of the reference list of the citing article. If the reference list was not accessible, the metadata listed on Crossref for the invalid DOI was compared to the metadata of the article resolved with the cleaned DOI. The purpose of this step is to measure the overall precision of the conceived algorithm. Three cases were differentiated: 1. the reference to the cleaned DOI was verified, 2. no reference to the cleaned DOI could be found, and 3. the reference list and metadata of the given articles were not accessible, making it impossible to produce a statement about the correctness of the DOI.

## Results

Regarding the first research question about the classes of DOI errors, the taxonomy described by Xu et al. (2019) was adopted, as can be seen in Figure 1. As described in Figure 3, by modifying their regular expressions, it was possible to adapt the regular expression for the prefix-type errors to be more efficient for the needs of this research. Also, automatic methods were implemented to clean up one additional pattern for the prefix-type errors, and nine additional patterns for the suffix-type errors. By applying these methods to the dataset, a part of the given DOI errors was fixed and the results were measured. The resulting CSV file has been published on Zenodo (Boente et al., 2021c).

The numbers can be examined here:

| ***Set of DOIs*** | **# Occurrences** |
|---|---|
| Number of invalid DOIs after cleaning | 1,024,275 |
| Number of valid DOIs after cleaning | 199,020 |

*Table 7 Report on the number of DOI names cleaned through the proposed algorithm.*

About 16% of the whole dataset were cleaned by the methods presented in this study. In the following section, the distribution between the different subclasses of errors in the dataset of valid DOIs is shown to differentiate the impact of the distinct methods.

| ***Set of DOIs*** | ***# Occurrences*** |
|---|---|
| Suffix errors fixed | 119,146 |
| DOIs temporarily invalid now valid | 57,196 |
| Prefix errors fixed | 21,742 |
| Other-type errors fixed | 953 |

*Table 8 Report on the number of DOI names belonging to the different classes that have been fixed.*

In Table 8, it becomes evident that the DOI names made valid by fixing a suffix error constitute more than half of the set of valid DOI names after the application of our algorithm. The next biggest set of valid DOIs consists of the DOI names that were only temporarily invalid and marked as valid before applying any cleaning method. The prefix errors compose around 11% of the valid DOIs, and other-type errors could only be found rarely.

To compare the extended methods used in this work to the ones introduced by Xu et al. (2019), the outcomes of their regular expressions can be observed separately in figure 5. It must be said that it was not possible to reproduce their procedure exactly, as the source code used in their study was not made available. As a consequence, the comparison has been made by using their set of regular expressions in our pipeline and by checking how many of the DOIs cleaned with them were returned as valid by the DOI proxy REST API.

The results can be made visible using a bar chart, which helps to compare the values through the difference in length of the associated bars and to confront the numbers resulting from the regular expressions from Xu et al. (2019).

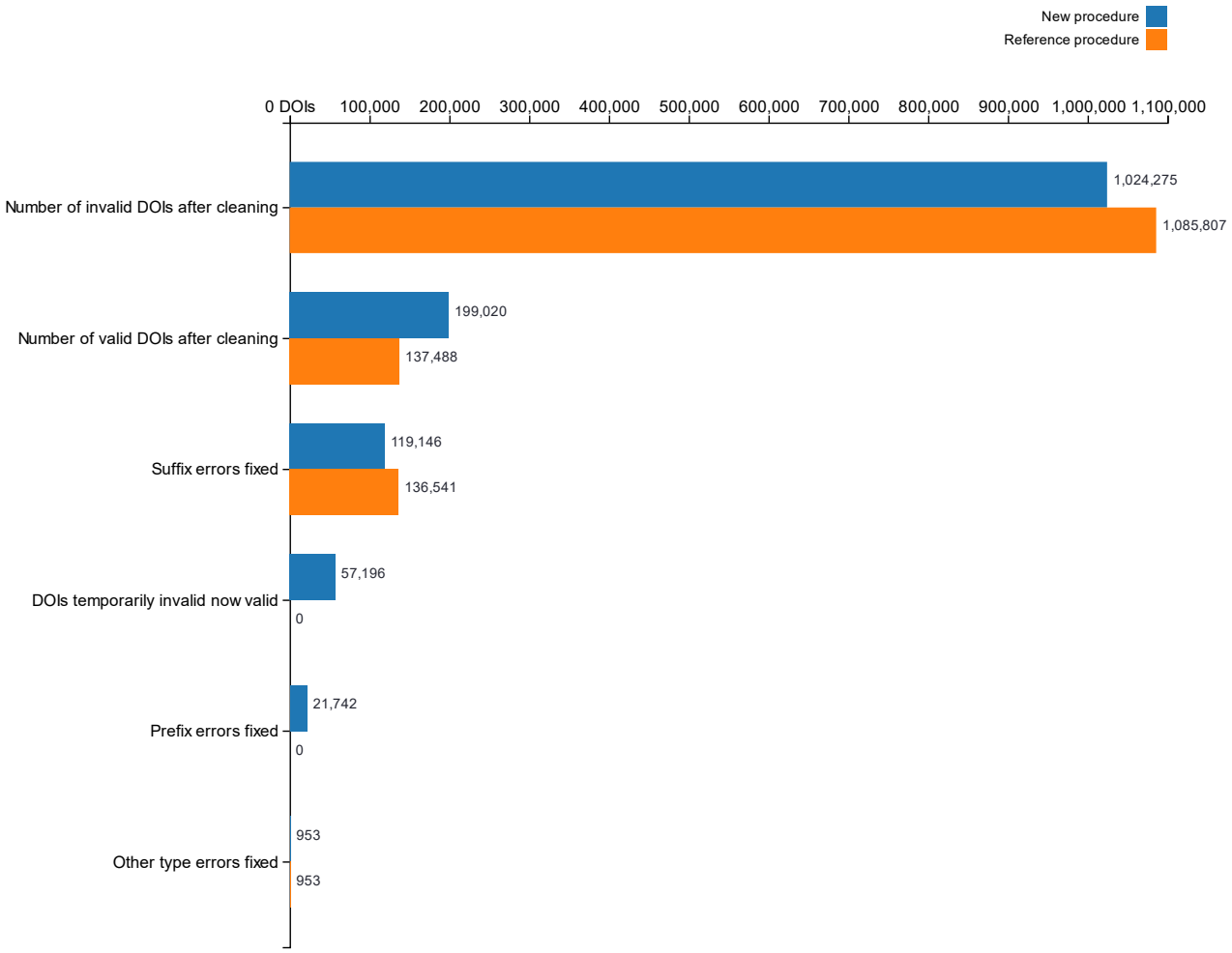


*Figure 5 Bar chart showing the results obtained by the reference procedure described by Xu et al. (orange) in comparison with the extended procedure described in this research article (blue).*

An alternative visualization using a treemap helps to read the data created in the study described here from a hierarchical point of view and to compare the values through area differences. In this case, the root is composed of the set of all DOIs, which includes the DOIs still invalid after the algorithm has been run and those that have been corrected.


*Figure 6 Treemap showing the results obtained.*

Both visualizations were created using d3.js v6.7.0 (Bostock, 2021), and it is possible to interact with them at the following address: https://open-sci.github.io/2020-2021-grasshoppers-code/.

The manual investigation of a subset of 100 cleaned DOI names can be found in the source code at "output/100_random_cleaned_dois.csv" (Massari et al., 2021). In the 100 randomly chosen cleaned DOIs, 98 proved as correct, one proved as incorrect, and one other was not accessible.

## Discussion and Conclusions

Applying regular expressions to a dataset of invalid DOI names can clean and identify different classes of factual errors, defined as subclasses of prefix-, suffix-, and other-type errors. Making around 16% of the given dataset valid, the methods applied in this paper can be seen as successful. This 16% also include a large number of DOIs that became valid over time without applying any cleaning method. Subtracting these, it can be stated that around 11% of the dataset were made valid using cleaning methods.

The largest part of these DOIs was cleaned from suffix-type errors, supposing that these types of errors are the most prominent in this set of invalid DOIs. On the other hand, it should be noted that the suffix-cleaning regular expressions that were used are also broader than the ones used for the other types of errors. Including 9 different

options, the higher matching rate can be seen as expectable, and it does not necessarily reflect on any characteristic of the given dataset.

98% of the cleaned cited DOIs proved as correct after the manual check of the randomly generated subset of 100 DOIs. However, a closer investigation of the two DOIs resulted as incorrect showed that the reason is not to be imputed to the algorithm adopted. The DOI "10.1007/978-3-319-90698-0_26" (Krebs, 2018) should appear among the references of the article corresponding to the DOI "10.17660/actahortic.2020.1288.20" (Wang et al., 2020), which unfortunately is behind a paywall and therefore inaccessible. This article was not the only one published on a toll-access journal among the 100 checked DOIs, but the University of Bologna, with which the authors of this research are affiliated, is not subscribed to the magazine in question, namely "ISHS Acta Horticulturae". On the other hand, the DOI "10.1007/s10479-011-0841-3" (García-Alonso, 2014) should be cited by "10.1101/539833" (Domanskyi et al., 2019), but, although reported by Crossref, it does not appear in the original article, not only in its correct version but also in the invalid one, that is "10.1007/s10479-011-0841-3." with a point after the DOI. This mismatch can be explained as a processing error made by the platform that published the article, that is bioRxiv, since the same invalid DOI appears 118 times in the dataset analyzed and not even once in the articles that mention it.

The comparison to the reference study made by Xu et al. (2019), as can be seen in Figure 5, shows that the methods used in this research were able to clean more DOIs overall. However, which could be seen as surprising, applying the methods from Xu et al. allowed 17,395 more suffix errors to be cleaned. On the other hand, the reference procedure did not clean any prefix errors in the given dataset. This result can be explained by taking a closer look at the regular expressions and the definition of prefix- and suffix-type errors used by Xu et al. In the study presented here, the regular expression for prefix errors was broader, as it considered prefix errors any additional strings before the DOI, even if they were not at the beginning of the string. On the other hand, the reference procedure considered prefix errors only those at the beginning of the string, hence the absence of matches in the given dataset. The current study, for instance, in the DOI "10.1016/J.JLUMIN.2004.10.018.HTTP://DX.DOI.ORG/10.1016/J.JLUMIN.2004.10.018", considered the presence of the DOI proxy server as a prefix-error, because it appeared before the DOI. The higher number of suffix errors cleaned with the methods by Xu et al. comes as a consequence. Since the present algorithm subsequently checks a DOI name for prefix, suffix and, finally, other-type errors, the errors not caught as prefix errors are cleaned as suffix errors in the next step. Presumably, the mistake related to the presence of the DOI proxy server was fixed as a suffix error in the Xu et al. procedure, which led to a higher number of matches.

This difference of the studies also makes evident that defining classes of errors is a task open for interpretation and seemingly small decisions can lead to different results. To define regular expressions in the first place, a manual investigation of the dataset is required. This implies that they depend on the given data and the researcher extracting the information. Therefore, the regular expressions used in this study can only represent and clean a part of the existing errors in the data.

As mentioned in the *Materials and Methods* section, working with a dataset of more than one million invalid DOI names was resource-intensive and almost brought this work's resources to their limits. Having the goal to check all DOIs for validity before cleaning and verifying the results by a second validation after the cleanup, running the whole dataset using the DOI proxy server took more than a week. The alternative, checking DOI names for validity using the work level data from Crossref, led to high RAM consumption, and thus, it was not fit for this research.

To be used on bigger datasets in the future, validation methods for DOI names should be evolved to consume fewer resources, or the given algorithms could be run on a machine with a higher RAM capacity. This work can still contribute to an improvement in the data quality in the field of Scientometrics, extending the taxonomy of DOI name errors and defining further elaborated regular expressions fit to clean a respectable amount of those errors.

## References


- Boente, R., Massari, A., Santini, C., & Tural, D. (2021a). Classes of errors in DOI names (Data Management Plan) (Version 5). Zenodo. https://doi.org/10.5281/zenodo.4733919
- Boente, R., Massari, A., Santini, C., & Tural, D. (2021b). Protocol: Investigating DOIs classes of errors. protocols.io.
- https://dx.doi.org/10.17504/protocols.io.buuknwuw
- Boente, R., Massari, A., Santini, C., & Tural, D. (2021). Classes of errors in DOI names: output dataset (Version v1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4892551
- Bostock, M. (2021). D3: Data-Driven Documents. Software Heritage. https://archive.softwareheritage.org/swh:1:dir:35fe697ae5a21e96d9fc01d890b30010e23c16dd
- Buchanan, R. A. (2006). Accuracy of cited references: The role of citation databases. College and Research Libraries, 67(4), 292–303. https://doi.org/10.5860/crl.67.4.292
- Cioffi, A., Coppini, S., Moretti, A., & Shahidzadeh A.N. (2021, May 3). Investigating missing citations in COCI and publishers involved (Version First). Zenodo. http://doi.org/10.5281/zenodo.4735636
- Crossref. (2021). January 2021 Public Data File from Crossref. https://doi.org/10.13003/GU3DQMJVG4
- Domanskyi, S., Szedlak, A., Hawkins, N. T., Wang, J., Paternostro, G., Piermarocchi, C. (2019). bioRxiv 539833. https://doi.org/10.1101/539833
- Franceschini, F., Maisano, D., & Mastrogiacomo, L. (2015). Errors in DOI indexing by bibliometric databases. Scientometrics, 102(3), 2181–2186. https://doi.org/10.1007/s11192-014-1503-4
- García-Alonso, C.R., Pérez-Naranjo, L.M. & Fernández-Caballero, J.C. (2014). Multiobjective evolutionary algorithms to identify highly autocorrelated areas: the case of spatial distribution in financially compromised farms. Ann Oper Res 219, 187–202. https://doi.org/10.1007/s10479-011-0841-3
- Heibi, I., Peroni, S., & Shotton, D. (2019). Software review: COCI, the OpenCitations Index of Crossref open DOI-to-DOI citations. Scientometrics, 121(2), 1213–1228. https://doi.org/10.1007/s11192-019-03217-6
- International DOI Foundation. (2019). DOI® Handbook. https://doi.org/10.1000/182
- Krebs, S.L. (2018) Rhododendron. In: Van Huylenbroeck J. (eds) Ornamental Crops. Handbook of Plant Breeding, vol 11. Springer, Cham. https://doi.org/10.1007/978-3-319-90698-0_26
- Massari, A., Santini, C., & Boente, R. (2021). open-sci/2020-2021-grasshoppers-code: Classes of errors in DOI names (Version 1.1.0). Zenodo. https://doi.org/10.5281/zenodo.4723983
- Peroni, S. (2021). Citations to invalid DOI-identified entities obtained from processing DOI-to-DOI citations to add in COCI [Data set]. Zenodo. https://doi.org/10.5281/zenodo.4625300
- Wang, S., Van Huylenbroeck, J. and Zhang, L.-H. (2020). Adaptability of Rhododendron species to climate and growth conditions at Lushan Botanical Garden. Acta Hortic. 1288, 131-138. https://doi.org/10.17660/ActaHortic.2020.1288.20
- Xu, S., Hao, L., An, X., Zhai, D., & Pang, H. (2019). Types of DOI errors of cited references in Web of Science with a cleaning method. Scientometrics, 120(3), 1427–1437. https://doi.org/10.1007/s11192-019-03162-4
- Zhu, J., Hu, G. & Liu, W. DOI errors and possible solutions for Web of Science. Scientometrics 118, 709–718 (2019). https://doi.org/10.1007/s11192-018-2980-7