Journal of Integrated Information Management

Vol 8, No 2 (2023)

Jul-Dec 2023

Open and closed citations' indexes results: a comparative study

Stefano Sorrentino

To cite this article:

Sorrentino, S. (2023). Open and closed citations' indexes results: a comparative study. *Journal of Integrated Information Management*, *8*(2), 7–13. Retrieved from https://ejournals.epublishing.ekt.gr/index.php/jiim/article/view/38144

Journal of Integrated Information Management electronic journal

Open and closed citations' indexes results: a comparative study

Stefano Sorrentino
University of Bologna
stefano.sorrentino4@studio.unibo.it [**ORCID:** 0009-0001-9166-9396]

Article Info

Article history:
Received 20 October 2023
Received in revised form 21 November 2023
Accepted 02 December 2023

http://dx.doi.org/10.26265/jiim.v8i2.38144

Abstract:

Purpose *- The following paper is a comparative study of the differences in the results provided by different academic and scholar indexes regarding a sample of DOI-identified articles and papers: with citation metrics being more and more relevant for the evaluation of scholars and their works, it is crucial to understand the differences between different indexes, their results and their functioning, digging into both open and close scenarios.*

Design *- The results of four different indexes (Elsevier's Scopus, Google Scholar, Dimensions, OpenCitations) have been compared through the provided REST APIs, when possible, and Python web scraping libraries. Different features have been considered for drawing the results, such as the easiness for the user to retrieve such metrics and their metadata and the reasons behind the differences in the results.*

Findings *- The study highlights the advantages of open citation metrics indexes and Linked Open Data for the final user. Still, at the same time, it points out how, when it comes to the completeness of the results, traditional indexes still provide more in-depth coverage of the academic literature, identifying the need to keep working to integrate more indexes and sources in the open ecosystem.*

Originality/value *- This study aims to call attention to the strengths and advantages of FAIR approaches in the field of citation metrics, providing a successful example of an open alternative to traditional indexes.*

Index Terms — citations – indexes – FAIR – metrics – LOD

I. Introduction

Regarding citation metrics, it is crucial to understand the reasons behind the differences in the results provided by different indexes and the criteria used to rank scholars, researchers, and their works.

Historically, the landscape has been shaped by a predominance of metrics retrieved from closed indexes managed by commercial publishers, which often don't share the citation data with open environments. In recent years, though, with the advent of open science practices and FAIR principles, efforts have been made to propose an open approach to citation metrics. In this context, the Initiative for Open Citations (I4OC) **[1]** pushes for the availability of data on citations that are structured, separable, and open, offering a disrupting alternative to the predominant scenario composed mainly of subscription-based indexes managed by commercial organizations.

Among the founders of the I4OC is OpenCitations, "an infrastructure organization for open scholarship dedicated to the publication of open citation data as Linked Open Data using Semantic Web technologies" **[2]** managed by the Research Center for Open Scholarly Metadata at the University of Bologna. Since its birth in 2010, it has configured itself as an alternative to traditional scholarship indexes and organizations, both from a technological and ethical point of view.

The non-openness of the references contained in the vast majority of publications leads to difficulty in retrieving metadata from open indexes, which is often the result of combined different causes: either because publishers won't submit to platforms such as Crossref (the DOI provider on which OpenCitations relies the most) the references of their publications, or because they have obtained their DOI through a different organization or, finally, because they publish in plain text/PDF format, preventing the occurrence of the publication in any infrastructure that relies on machine-readable formats. OpenCitations' data model heavily relies on Linked Open Data, the Resource Description Framework (RDF) and the semantic web: this allows to treat citations as first-class data entities, hence with a unique identifier (Open Citation Identifier, OCI), and to convey metadata about the citation itself (which is different from the bibliographic metadata of the citing and cited entity).

In 2022, the Open Citations "Index of Crossref Open DOI-to-DOI Citations" (COCI) reached the number of about 1.3 billion citation records **[3]**, that is to say, 52% of what is provided by Google Scholar, vs the slightly greater 58% of Elsevier's Scopus in comparison. Since then, Open Citations has expanded its indexes with the addition of other sources, such as DataCite, NIH Open Citation Collection, OpenAIRE and Japan Link Center (JaLC). Furthermore, Open Citations allows third parties to submit citation data concerning their publications to fill the gap of the missing citations from some of the biggest publishers not available in Crossref as open material (Elsevier being the main one).

II. Tools

A first comparison between different indexes (in this case OpenCitations, Google Scholar, Elsevier's Scopus, and Dimensions) can be carried out by analyzing the tools they provide to the users to retrieve citation data about bibliographic resources.

Starting from OpenCitations, all the triples that describe the entities, attributes and relations are stored in a triplestore database and can be queried using the SPARQL language. The result would be the set of OCIs that identify the Citations entities with the bibliographic resource identified by the queried DOI as a cited entity, each of which can be explored in the RDF/XML, Turtle, or JSON-LD format.

Furthermore, to allow users who are not experts in the use of the SPARQL language to query the dataset, Open Citations provides, besides a search interface on the website, two REST APIs (respectively for the "Index" **[4]** and "Meta" **[5]** dataset): these have been made available thanks to the development of an open-source software, RAMOSE (Restful API Manager Over Sparql Endpoints) **[6]**. Like the whole data model itself, RAMOSE can be used by developers of any application to provide REST APIs over a triplestore.

Concerning Google Scholar (which indexes metadata of scholarly literature across a vast array of disciplines and publishers), the service per se doesn't provide a way to automatically retrieve data, such as a "Google Scholar API", but independent developers and users have developed tools to do so: in this case, SERP API **[7]** has been used, which allows to extract from "Search Engine Results Pages" various kinds of information, including citations metadata from Google Scholar results.

On the other hand, Elsevier's Scopus, which is a leader in paid services when it comes to citation analysis tools and which includes peer-reviewed publications and metadata from a vast range of publishers, does provide an API to interact with their datasets, but with a paywall that prevents the user to freely extract certain kinds of data, such as citations metadata **[8]**.

The University of West Attica has a subscription to Elsevier's Scopus API, which allows one to visualize the number of citations of bibliographic documents on the UniWaCRIS webpage and which links to the Scopus webpage of that entry. Nevertheless, since, as we'll see, the considered dataset for this project is relatively small and focused on a specific field, to speed up the process, the retrieval of such information has been performed through means of web scraping, making use of the "Beautiful Soup" and "Selenium" python libraries.

Finally, UniWa also has an agreement with Dimensions, a relatively newer service in this landscape but which still indexes metrics concerning a vast range of bibliographic resources: to lean towards open access, a significant portion of its content is free of charge, but still some content is protected by paywalls. For this reason, only the number of citations per bibliographic entry retrieved from UNIWACRIS (https://uniwacris.uniwa.gr/) has been used for this study **[9]**.

III. Methods

The sample dataset used for this study, in .xls format, comes from the UniWaCRIS infrastructure, and it includes records from 879 bibliographic resources describing their metadata, such as the internal "id", the "collection id" and a list of dc-terms fields covering attributes such as the abstract, the responsible agents, the provenance metadata, the type of bibliographic resource and, of course, the identifiers. Among the various identifiers (DOI, ISBN, ScopusID, URL, etc.), DOIs have been chosen as the reference ones: this excluded all the entries that didn't have a DOI, reducing the number of considered resources to 303.

The developed Python software (which is available for consultation and reuse at this link: https://github.com/SleepingSteven/citations-analysis) is composed of different modules that address the following questions:

1. For how many DOIs does Open Citations provide the highest "cited by" value?
2. For how many DOIs does Elsevier's Scopus provide the highest "cited by" value?
3. For how many DOIs does Google Scholar provide the highest "cited by" value?
4. For how many DOIs does Dimensions provide the highest "cited by" value?
5. What is the average difference in the number of citations when using Scopus compared to when using Open Citations?
6. Comparing the results of Open Citations and Scopus, what are the differences in the citation results for each entry? Which citations don't appear, respectively?
7. What is the publisher of each missing citation?
8. What are the most common publishers of the missing citations for Open Citations and Scopus?

Concerning the first five points, the way in which the data have been retrieved differs depending on the index.
Starting from Open Citations, the file that performs the action is "resultsoc.py":
Source code

```
doislist=list()
dois = pd.read_excel('INSERT FILE NAME(e.g. filtered.xlsx)', usecols = 'COLUMN LETTER')
for i, row in dois.iterrows():

        doislist.append(row["dc.identifier.doi[en_US]"])

listaocindex=list()
for i in doislist:
    API_CALL = "https://opencitations.net/index/api/v2/citation-count/doi:"+i
    HTTP_HEADERS = {"accept": "application/json"}

    try:
        listaocindex.append(int(get(API_CALL, headers=HTTP_HEADERS).json()[0]["count"]))
    except Exception as e:
        listaocindex.append(0)

listaocindex.insert(0,"cited_by_oc")
workbook = xlsxwriter.Workbook('INSERT EXCEL OUTPUT FILE NAME HERE')
worksheet1 = workbook.add_worksheet()
worksheet1.write_column('A1', listaocindex)
workbook.close()
print (listaocindex)
```

The algorithm first retrieves all the DOIs from the filtered

Excel file through the "read_excel()" method of the "Pandas" Python library. Then, it proceeds to call the "Index" dataset REST API with the "/citation-count/" operation, providing as an argument each one of the DOIs through the ".get()" method of the "Requests" Python library. The result is a .json output (whose format was specified in the HTTP headers) with a single object and with a single key-value association: "count": "number of citations".

e.g.:

```
[
    {
"count": "5"
    }
]
```

The retrieved list of "cited by" values is then loaded into an output Excel file through the "xlsxwriter" Python library. It is then ready to be further treated (in this case, being added as a new column with the name "cited_by_oc" to the "filtered.xslx" file, available on the GitHub page of the project).

For what concerns Scopus, what follows is a section of the "resultsscopus.py" file, which was written keeping in mind what is stated in the "tools" section concerning Scopus API:

Source code

```
    # Check if the request was successful (status code 200)
    if response.status_code == 200:
        # Parse the HTML content of the webpage
        soup = BeautifulSoup(response.text, 'html.parser')

        # Extract the value based on HTML structure and tags
        try:
            value_to_extract = soup.find('div', class_="metric-counter-scopus").text
            listascopusindex.append(int(value_to_extract.replace('\t', '').replace('\n',
''')))
            # Print the extracted value
            print(int(value_to_extract.replace('\t', '').replace('\n', '')))
        except Exception as e:
            listascopusindex.append(0)


    else:
        print(f"Failed to retrieve the webpage. Status Code: {response.status_code}")

print (listascopusindex)
listascopusindex.insert(0,"cited_by_scopus")
workbook = xlsxwriter.Workbook('INSERT EXCEL OUTPUT FILE NAME HERE')
worksheet1 = workbook.add_worksheet()
worksheet1.write_column('A1', listascopusindex)
workbook.close()
```

Slightly differently to the previous point, the first thing to do was retrieve the UNIWACRIS URIs (instead of the DOIs) from the Excel file: with them it was possible to generate HTTP get requests and to obtain the .html content of the web page, from which it was possible, thanks to the "BeautifulSoup" library, to extract the section related to the Scopus banner, identified by the class="metric-counter-scopus".

After retrieving the values, they are again stored in an output file.

The algorithm for retrieving Dimensions' indexes in the "resultsdimensions.py" file is similar, with the change that also the "Selenium" Python library is used since a script generates the Dimensions' banner and the library allows to read the .html code dynamically:

Source code

```
        # Access the HTML content of the embedded element
        embedded_element_html = embedded_element.get_attribute('outerHTML')
        soup=BeautifulSoup(embedded_element_html, 'html.parser')
        try:
            value_to_extract = soup.find('div', class_="__db_score_normal").text
            listascopusindex.append(int(value_to_extract.replace('\t', '').replace('\n',
''')))
            print (int(value_to_extract.replace('\t', '').replace('\n', '')))
            print ("ok")
        except Exception as e:
            listascopusindex.append(0)
            print (0)
            print("no")


    finally:
        # Close the browser
        driver.quit()
```

Finally, coming to Google Scholar, to facilitate the use of the SerpAPI, the "SerpApiGoogleScholar" **[10]** Python library has been used to retrieve then the .json output, which included the number of citations for each queried DOI.

SerpApi, though, can be used to make 100 searches a month on its basic plan: so, for the remaining 204 entries left to be queried, another custom backend of the same "SerpApiGoogleScholar" library was used, which allows to retrieve data from Google Scholar without the need of an API but at a lower rate.

Here's the code from the "resultsgooglecustom.py" file:

Source code

```
for i in doilist:

    try:
        data = parser.scrape_google_scholar_organic_results(
            query=i,
            pagination=False,
            save_to_csv=False,
            save_to_json=False
        )
        try:
            citationcountd[i]=data[0]["cited_by_count"]
            citationcountl.append(data[0]["cited_by_count"])
```

Since, as expected and as it will be shown, Google Scholar results are greater than double the ones provided by the other three indexes, respectively, the focus was shifted exclusively on the comparison of Open Citations with Scopus and Dimensions, referring to Google Scholar only for further information retrieval.

Also, being Scopus cited multiple times in the Open Citations documentation as a "competitor" and being the automized retrieval of data much smoother when dealing with Scopus, the following analyses are focused solely on the differences with Elsevier's infrastructure.

Points 6, 7, and 8 have been addressed through the development of the "analysis.py" file: the algorithm first checks the differences in the citations count between Open Citations and Scopus through the previously retrieved data, to then proceed to retrieve the list of citations from both Open citations and Scopus, to find the ones appearing only in one of the two indexes.

Concerning Open Citations, the "Index" dataset API call "https://opencitations.net/index/api/v2/citations/doi: + DOI" allows retrieval in JSON format the list of Citation objects that have as a cited document the one identified by the submitted DOI.

After sliding the list of objects and retrieving the DOIs of the citing documents, the "Meta" dataset API call

"https://opencitations.net/meta/api/v1/metadata/doi: + DOI" can be used to retrieve bibliographic metadata about that document, including the name. This was needed because, as we will see, the comparison between the citations listed in Scopus and Open Citations had to be done by name: for the same reason, the names were uniformed and brought to lowercase.

The retrieval of the Scopus list of citing documents' names for each DOI was, on the other hand, once again impossible to perform through Elsevier's API on the basic developer plan. The result was nevertheless achieved through the development of the "find()" function: after setting up "Selenium" with the browser profiles, it was possible to retrieve the dynamically generated .html code of the Scopus "results" pages submitting at each iteration a get-request function having the following URL as argument: "https://www.scopus.com/results/citedbyresults.uri?sort= plf-f&cite= + SCOPUSID", where SCOPUSID is, for each of the excel entries, the value of the "dc.identifier.scopus;" column.

In fact, Scopus allows users to visualize information of the citing documents for a submitted Scopus-ID on a "read-only" page:

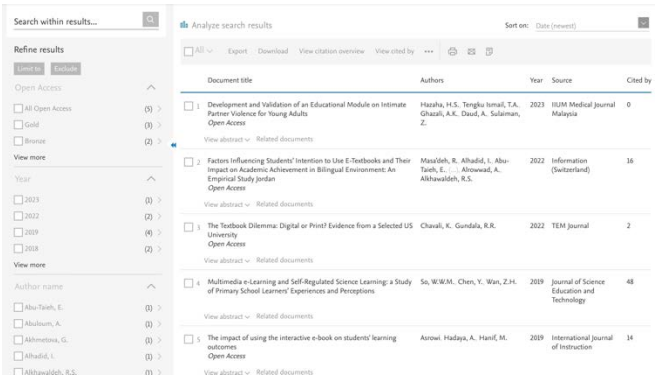

Fig 1. Scopus web page providing information of the citing documents

As it is visible from this screenshot, only the names of the citing documents are provided as a form of identification (the reason why the names were retrieved through Open Citations "Meta" API as previously explained): all of them are hence stored in a list for each iteration, thanks to the retrieval through "BeautifulSoup" of the HTML section concerning the elements under the class "docTitle".

Once the list of citing documents both from Scopus and Open citations for each entry is retrieved, the algorithm checks for missing elements in both and stores them in a list to recover their publication information:

Source code

```
print ("Difference for DOI: "+dois.loc[i]['dc.identifier.doi[en_US]']+": ")
        print(result)
        for i in result:
            successful=False
            while not successful:
                try:
                    parser = CustomGoogleScholarOrganic()
                    data = parser.scrape_google_scholar_organic_results(
                        query=i,
                        pagination=False,
                        save_to_csv=False,
                        save_to_json=False
                    )
                    if "publication_info" in data[0]:
                        try:
                            print((data[0]["publication_info"].split(" - "))[1])
                            successful=True

publishersnotinscopus.append((data[0]["publication_info"].split(" - "))[1])
                            print (publishersnotinscopus)
                        except Exception as b:
                            successful=True
                            print("Ok but formatting error")
                    elif data:
                            successful=True
                            print("Ok but formatting error")
                except Exception as e:
```

These instructions obtain information from Google Scholar through the previously used "Custom Google Scholar" backend, including the information data of each citing publication.

The information is retrieved once again in the form of a "key-value" pair in a list of JSON objects, which usually follows this pattern:

```
[
  {
   ...
   "publication-info": "The Electronic Library, 2020 - emerald.com"
   ...
   }
]
```

The first part of the string identifies the publication venue, and the second one, divided by a dash, is the publisher.

The goal was then to obtain, for both Open Citations and Scopus, the list of publishers of the citing documents appearing (and missing) exclusively in their results, also using the "countpublishers()" function to obtain a percentage of how many times a specific publisher is present.

Finally, the "counttypes()" function counts the percentage of the type of documents (conference papers or articles) that present a higher "cited-by" count either in Open Citations or Scopus.

IV. RESULTS

The results are depicted in the following figures:

• Total number of citations with the provided DOIs as cited document according to different indexes:

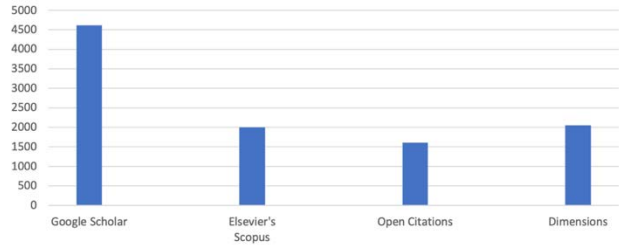

Fig 2. Total number of citations with the provided DOIs as cited document according to different indexes

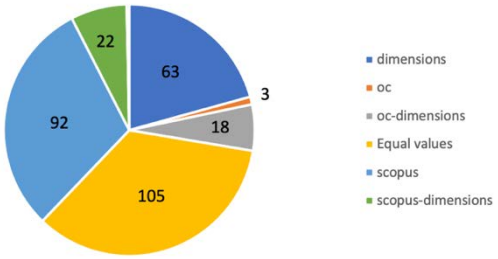

Fig 3. DOIs for which different indexes provided the highest "cited-by" value (excluding Google Scholar)

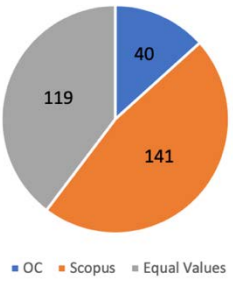

Fig 4. DOIs for which different indexes provided the highest "cited-by" value (considering only Open Citations and Scopus)

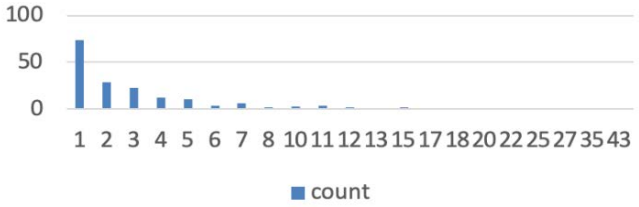

Fig 5. Average difference in the number of citations per DOI between Open Citations and Scopus (2,15, standard deviation = 4,89)

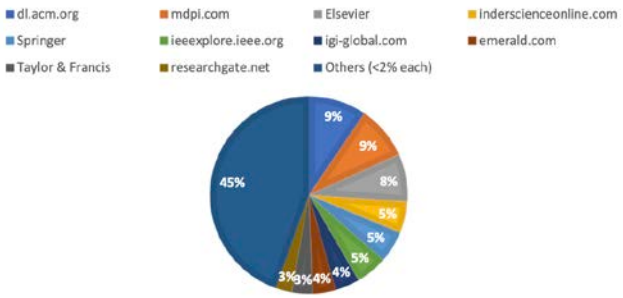

Fig 6. Publishers of citing documents listed by Scopus and not by Open Citations

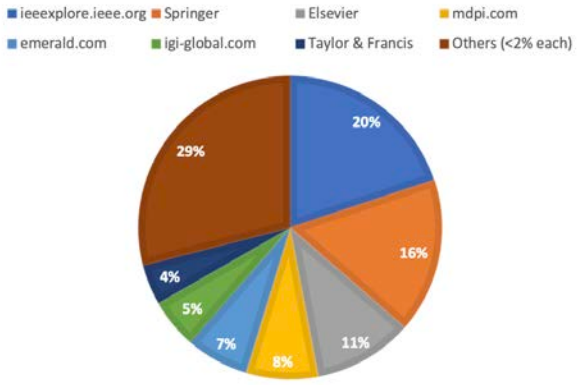

Fig 7. Publishers of citing documents listed by Open Citations and not by Scopus

The results seem to have confirmed, in the first place, the numbers proposed by the Open Citations' documentation, with Google Scholar providing a total number of citations twice as large as those offered by the other indices.

The reason for this might lie behind the Google Scholar indexing criteria, which guarantees comprehensive coverage, indexing a diverse array of sources such as preprints, conference papers, and institutional repositories.

As a matter of fact, being listed on Google Scholar for a citation entry is a smoother process since the infrastructure relies on powerful crawling technologies (similar to the ones used by the same Google Search engine) and in-text citation recognition, which make it easier even for independent or smaller publishers to be listed among the results. There are just a few prerequisites, like the documents being readable in at least .pdf or .html format, being of course, scholarly articles, and divided into sections (abstract, title, author, references, bibliography): if these features are matched, and if the websites on which the documents are hosted do not present anti-crawling features or do not use uncommon protocols, the bibliographic and citation metadata will most likely appear on Google Scholar **[11]**.

On the contrary, Elsevier's Scopus (as well as other academic indices) focuses more on selecting peer-reviewed works and submitting academic titles, which is subject to stricter selection standards and a longer process.

Also, Scopus evaluates annually the performance of every work within its database. Each work must meet specific citation metrics and benchmarks: should a journal fall short of these benchmarks for two consecutive years, it would be flagged for re-evaluation, potentially leading to its removal from the corpus **[12]**.

When looking at the direct comparison between Scopus and Open Citations, we observe that for 141 DOIs, Scopus provided more citation results, versus the 40 where Open Citations turned out to be more comprehensive, with an average difference of 2,15 citations per document.

Digging into the results, it is observed that among the citations which were "exclusive" to Scopus, 55% of them referred to the publisher of the citing document, a well-known one, but with a maximum individual percentage of 9% of the results (the top ones being ACM - Association for Computing Machinery publishing, MDPI, Elsevier, Springer,

IEEE, Emerald, Taylor & Francis). The remaining 45% was represented by publishers who published less than 2% of the citing documents, and the reason might be that they were mostly smaller independent ones, universities, specific repositories, etc.

On the other hand, the same well-known publishers represented 71% of the publishers of citations being listed only by Open Citations, also presenting higher individual percentages (e.g., 20% for IEEE, 16% for Springer), with the "smaller ones" representing only 29%.

There could be many reasons for these disparities. For those that appear only in Scopus, looking at how the percentages tend to have fewer peaks throughout the results compared to the Open Citations ones, the reason might lie behind the general functioning of Open Citations, which relies mostly on publishers submitting the citation metadata of their publications to Crossref, the primary source of Open citations indexes **[13]**. At the same time, Scopus, as we've seen, is built upon the publishers' submission of peer-reviewed works to the platform. This might be seen as a priority by both smaller and bigger publishers in this case, compared to the submission of open metadata to a platform such as Crossref (which may also not be the provider of the DOI of the document), given the advantages that the listing of work in the Scopus network might bring to a publisher in terms of visibility.

Also, this kind of citations-metadata submission may not even be considered a required step in the publication flow and be ignored by smaller publishers who only include them as plain text, the reason why their percentage might be so high compared to the Open Citations results (45% vs 29%).

Concerning publishers of citing works appearing only in Open Citations, the less smooth percentage distribution, with peaks of 20%, 16% and 11% for single "big" publishers (respectively IEEE, Springer and Elsevier), and the smaller percentage of "little" ones, might imply in the first place the presence of citing documents whose publishers are accustomed to the good practice of submitting citation metadata to open platforms (or to platforms which agreed to provide their citations indexes to Open Citations for inclusion in their platform).

On the other hand, it may also imply the presence of works that were not accepted by Scopus (either because not peer-reviewed, because of low relevance, or because they didn't keep up with the recurring benchmark checks). Lastly, they may also not have been submitted to Scopus in the first place.

V. Discussion

The results match the ones proposed by a previous similar study that compared Elsevier's Scopus with other indexes, including Crossref, from which "57% of the citation links in Scopus cannot be obtained" **[14]** for reasons compatible with the above listed.

At the same time, though, the technical advantages of relying on Open Citations should be evident, at least when it comes to the data retrieval operations: the semantic Linked Open Data infrastructure, on which the whole infrastructure is built, ensures a faster, smoother, customizable, and cheaper process in comparison to other paywalled indexes, reason why the RDF structure and the ethics that support Open Citations data model should be taken into account and not be underestimated when it comes to a direct comparison with more comprehensive indexes.

Finally, this study suggests a reflection on the use of citation-based metrics as the sole indicator of a work's impact: restricted access to citation data due to paywalls and limited accessibility does not align with FAIR principles. In fact, such restrictions pose a threat to the transparency, replicability, and verifiability of research assessment, and data such as citation-based metrics may open up to all kinds of peculiarities and all kinds of issues may arise when collecting the related information (e.g. the different periods in which citations are accumulated and the related availability of such citations, the time that passes between a work and its first citation which affects the h-index, the "strategic" use of citations from the scientific community to gain advantage by citation-based metrics etc.).

For these reasons, Open Citations is working toward new implementations that guarantee more in-depth coverage of the academic literature. This means expanding its coverage to encompass references from publications using non-Crossref DOIs, references extracted from PDF documents, references provided by preprint repositories, and references related to data citations, views, savings, online discussions, and other non-textual research outputs to offer "altmetrics" **[15]** capable of monitoring impact beyond the academic landscape.

VI. References

[1] I4OC, "Initiative for Open Citations," https://i4oc.org/

[2] Silvio Peroni and David Shotton, "OpenCitations, an Infrastructure Organization for Open Scholarship," Quantitative Science Studies 1, no. 1 (February 2020): 428–44, https://doi.org/10.1162/qss_a_00023

[3] Silvio Peroni and David Shotton, "OpenCitations: Present Status and Future Plans," July 29, 2022, https://doi.org/10.5281/ZENODO.6940918

[4] "OpenCitations - OpenCitations Index,", https://opencitations.net/index

[5] "OpenCitations - OpenCitations Meta,", https://opencitations.net/meta

[6] "Opencitations/Ramose," Python (2018; repr., OpenCitations, June 7, 2023), https://github.com/opencitations/ramose

[7] "SerpAPI", https://serpapi.com

[8] Elsevier Developer Portal", https://dev.elsevier.com

[9] Kouis, D., Veranis, G., Zervas, M., Artemis, P., Giannakopoulos, A., Bellas, C., & Christopoulou, K. (2021). Citation indexes integrated management for Institutional Repositories data enrichment. Journal of Integrated Information Management, 6(1), 14–24. https://doi.org/10.26265/jiim.v6i1.4490

[10] Dmitiry Zub, "Dimitryzub/Scrape-Google-Scholar-Py," Python, January 8, 2024, https://github.com/dimitryzub/scrape-google-scholar-py

[11] "Google Scholar Help," 2024, https://scholar.google.com/intl/it/scholar/inclusion.html#overview
[12] "Scopus Content Policy and Selection | Elsevier,", https://www.elsevier.com/products/scopus/content/content-policy-and-selection
[13] Ivan Heibi, Silvio Peroni, and David Shotton, "Software Review: COCI, the OpenCitations Index of Crossref Open DOI-to-DOI Citations," Scientometrics 121, no. 2 (November 2019): 1213–28, https://doi.org/10.1007/s11192-019- 03217-6
[14] Martijn Visser, Nees Jan van Eck, and Ludo Waltman, "Large-Scale Comparison of Bibliographic Data Sources: Scopus, Web of Science, Dimensions, Crossref, and Microsoft Academic," Quantitative Science Studies 2, no. 1 (April 8, 2021): 20–41, https://doi.org/10.1162/qss_a_00112
[15] "Altmetrics: A Manifesto – Altmetrics.Org,", 2024, http://altmetrics.org/manifesto/

VII. AUTHORS

Stefano Sorrentino, after obtaining his bachelor's degree in Humanities in 2022, he is now a graduating master' student at the University of Bologna. He has been an Erasmus+ visiting student at the "Archival, Library and Information Studies" department at the University of West Attica in 2023, and he is currently working on his final data science thesis project in Digital Humanities and Digital Knowledge at CERN, in Geneva, as an Administrative Student.