Creating Open Citation Data with BCite

Marilena Daquino[1], Ilaria Tiddi[2], Silvio Peroni[1]

[1] Department of Classic Philology and Italian Studies, University of Bologna, Italy
marilena.daquino2@unibo.it, silvio.peroni@unibo.it
[2] Knowledge Media Institute, The Open University, United Kingdom
ilaria.tiddi@open.ac.uk

Abstract. In the past year we have seen a huge release of open citation data, thanks to the effort of several parties such as the Initiative for Open Citations (I4OC). However, the *coverage* of such data is one of the most important issues that the open citation data community is currently facing. In this paper we present an approach to create open citation data while supporting journal/volume editors in the process of curating the reference lists contained in camera ready articles according to the particular style and format guidelines forced by the publishers. In particular, our contribution is twofold: (a) a basic workflow to support editors in the management and curation of their data and (b) a tool, called BCite, to create open citation data compliant with an existing RDF-based repository: the OpenCitations Corpus.

Keywords: Open Citation Data, OpenCitations, Citation Data Curation, Open Science, Semantic Publishing

1 Introduction

The Initiative for Open Citations (I4OC, `https://i4oc.org`) was launched in April 2017 to convince publishers, depositing their citation data in Crossref (`https://crossref.org`), to release them in the public domain as **structured** (i.e. expressed in machine-readable formats), **separable** (i.e. available without the need to access the source bibliographic products in which the citations are created) and **open** (i.e. freely accessible and reusable) data – **open citation data** from now on. Even if several of these data (i.e. about 500 million citation links from 19 million articles) have been released during the past year and are now available through the Crossref API (`https://api.crossref.org`), they are not exposed natively with Semantic Web technologies. To this end, several projects and organisations supporting I4OC, such as OpenCitations (`http://opencitations.net`) [11], work daily on providing open citation data in RDF so as to be queried by using SPARQL and to be accessed by the usual content negotiation mechanism.

However, the *coverage* of such data is one of the most important issues that the open citation data community is currently facing. This is due to at least two factors. On the one hand, some academic disciplines, in particular the Social Sciences and the Humanities, are inherently under-represented in such citation datasets. In fact,

bibliographic references included in articles of these disciplines tend to be missed even in major (non-open and commercial) citation indexes such as Scopus (`https://www.scopus.com/`) and Web of Science (`https://clarivate.com/products/web-of-science/`). For instance, according to our knowledge, while the article [3] (published in the *Conservation Science in Cultural Heritage* journal) references [12], the related citation is not reported in any citation index. On the other hand, several journals and/or publishers do not have the financial capabilities nor the technical support to submit their citation data to Crossref so as to expose them in the public domain as open citation data – and to be, then, easily reused by other parties.

Two particular research questions can be derived by analysing the aforementioned situation:

1. is it possible to develop a mechanism which implements a workflow that allow one (a publisher, a journal editor, a researcher) to increase the coverage of existing RDF-based open citation datasets?
2. how much additional data would the aforementioned mechanism allow one to add to the existing RDF-based open citation datasets?

A possible solution to the aforementioned questions is to develop or adopt easy-to-use interfaces to allow users to create these missing citation data and, at the same time, to support them in some curation-related task with bibliographic data. In particular, the specific use-case we have explored in our work has concerned the last phase of the publication process of an article, i.e. when the journal/volume editor has to curate the reference list of an accepted article according to specific requirements (the reference style, the selection of the information included, external links, etc.) so as to be published in the final version-of-record of the article.

To this end, we have developed a prototype tool called BCite, which is introduced in this paper. BCite is based on a *tit-for-tat* strategy. The intuition is that, while editors can obtain clean references from a citing article they have in hand, they can also provide curated data as a contribution to public citation corpora, such as the OpenCitations Corpus (OCC) [11] – i.e. the RDF dataset of open citation data maintained by OpenCitations. BCite is designed to provide a full workflow for citation discovery, allowing users to specify the references as provided by the authors of the article, to retrieve them in the required format and style, to double-check their correctness, and, finally, to create new open citation data according to the OpenCitations Data Model [9], so as to be possible (in the future) their integration in the OpenCitations Corpus. We have also preliminary evaluated BCite so as to understand if it can be used to answer to the aforementioned research questions. The outcomes of our experiments are encouraging and show that, even with a limited set of input documents, the current coverage of of citation data in the OpenCitations Corpus can be extended easily using the application.

The rest of the paper is structured as follows. In Section 2 we introduce all the materials and methods we have used for implementing BCite. In Section 3 we introduce BCite by briefly describing its components and the workflow it implements. In Section 4 we evaluate our tool by using the reference lists included in some published articles of different disciplines. In Section 5 we introduce some of the most important

related works in the area. Finally, in Section 6 we conclude the work sketching out some future works.

2 Material and Methods

OpenCitations is a scholarly infrastructure organisation dedicated to provision of open bibliographic citations and associated tools and services. The main work of OpenCitations is the creation and current expansion of the Open Citations Corpus (OCC) [11], an open repository of scholarly citation data made available under a Creative Commons public domain dedication, which provides in RDF accurate citation information (bibliographic references) harvested from the scholarly literature. These are described using the SPAR Ontologies [10] according to the OpenCitations Data Model [9], and are made freely available so that others may freely build upon, enhance and reuse them for any purpose, without restriction under copyright or database law. The contents of OCC can be explored by humans using an appropriate search interface and navigated by means of another browse interface. Programmatic access to the OCC is available either using its SPARQL endpoint or its REST API (coming Q3 2018). Metadata for individual bibliographic entities can be accessed via a simple Web form using their individual URIs (e.g. `https://w3id.org/oc/corpus/br/1`).

The ingestion of citation data into OCC is currently handled by two Python modules, the *Bibliographic Entries Extractor* (BEE) and the *SPAR Citation Indexer* (SPACIN). While BEE is responsible for the creation of JSON files containing reference lists of published articles, SPACIN processes each of these JSON files, retrieves metadata about all citing/cited articles by querying the Crossref and the ORCID APIs, and stores the generated RDF resources both in the file system (as JSON-LD files) and the OCC triplestore. Part fo these two Python modules have been reused for the development of BCite, so as to demand them the full creation of the open citation data compliant with the OpenCitations Data Model.

3 BCite: a bibliographic reference correction service

BCite is a Web application that (a) allows users to curate the list of bibliographic references cited in a soon-to-be-published article, and (b) creates simultaneously RDF-based open citation data compliant with the OpenCitations Data Model, so as to be integrable into the OpenCitations Corpus. From now on, we refer to an article and to the bibliographic references included in its reference list as *citing entity* and *cited references* respectively, while we call *cited entities* the structured data derived from the latter.

Components. BCite is developed as a Python Web application (using the web.py framework) that can be run on a local machine. It is available on GitHub at `https://github.com/opencitations/bcite`, and includes the following logical components: (i) the *BCite App*, a web interface for data entry and data curation; (ii) the *BCite triplestore*, a Blazegraph instance (`https://www.blazegraph.com`) for storing the generated RDF triples; (iii) the *BCite API*, which is responsible of using the

OpenCitations Python modules for creating open citation data and store them in the triplestore. The BCite workflow and its components are shown in Figure 1.

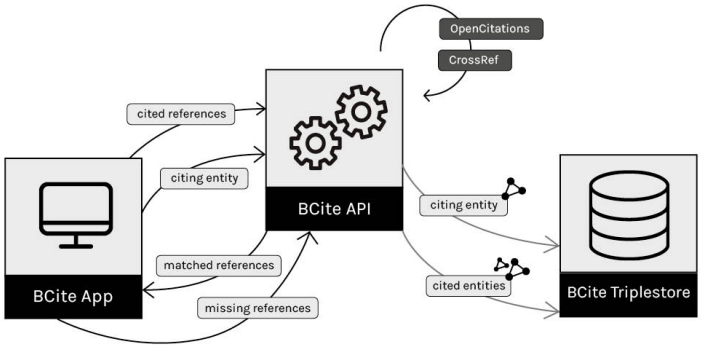

Fig. 1. BCite's workflow to clean bibliographic data and generate open RDF citations.

Workflow. BCite includes three main activities: the creation of a new citing entity, the lookup of its cited references, and the update of the triplestore.

Creating a citing entity. At first, the user inputs via the BCite App all the metadata of a citing entity (authors, title, journal, volume, issue, year, publisher, and DOI) and the list of cited references to be matched. A citation style can also be selected to format the output references. When submitting the metadata, a request is sent to the BCite API, which first creates a RDF representation of the citing entity, and then writes the generated triples in the BCite triplestore.

Reference lookup. For each of the cited references provided by the user, a request for matches in the BCite triplestore is sent to the BCite API. If no matches are found in the triplestore, the API (by means of the OpenCitations modules) looks in Crossref for getting structured additional metadata starting from the cited reference. For each of the matched references , an RDF resource (i.e. the cited entity) is either retrieved (if already present in the BCite triplestore) or created (if returned by Crossref). Then, the BCite API sends back to BCite App the reference text of the cited entity formatted according to the requested citation style. If Crossref does not return any match, an empty string is returned instead.

Correction and update. As shown in Figure 2, all the results are presented to the user in a table including (1) the submitted cited references in the first column, and (2) the proposed matched references in the second column. The user can: (a) accept the proposed reference text as is; (b) acknowledge that the proposed reference text refers to the intended cited article and modify some part of it; (c) reject the reference text when it does not depict the intended cited article and provide a new full reference text. Finally, all the accepted/corrected reference texts are sent back to the BCite API, and all the triples relevant to the citing and cited entities are stored

in the BCite triplestore, including the related provenance information according to the OpenCitations Data Model.

Fig. 2. The BCite App. The user inputs metadata about citing and cited entities, and obtains the bibliographic references formatted according to a selected citation style.

4 Evaluation

A preliminary evaluation of BCite was performed with the goal of demonstrating that our tool allows us to answer the two research questions mentioned in the introduction, i.e. to implement a mechanism to increase the coverage of existing RDF-based open citation datasets and to what extent, while facilitating the editors' workload during the data curation process of to-be-published journal articles. In particular, we were interested in (i) measuring the correctness of the references returned by the BCite API; and (ii) quantifying the contribution to the current status of the OpenCitations Corpus (OCC), i.e. how much new knowledge could, in principle, be added to it. To demonstrate these points, we use three cases different cases using journal articles from different disciplines.

1. A work [4] published by the *Journal of the Association for Information Science and Technology* (JASIST), that deposits all its bibliographic references in Crossref. In this scenario, aimed at assessing mostly the tool's precision, we modified the list of references to reproduce common mistakes that can be possibly introduced by either article authors or bibliographic curators (missing authors, wrong publication years or incomplete references, etc.). We therefore expect the precision of the tool to be high, and the contribution to OCC to be low.
2. A work [2] from the *Journal of Library and Information Science* (JLIS). While the journal's bibliographic data are curated and presumably indexed in major indices but not in Crossref, the paper belongs to the Digital Humanities area.

Hence, we expect a lower precision but a higher contribution to OCC (i.e. a lower number of matches found in OCC).
3. A work [3] from a minor journal, i.e. *Conservation Science in Cultural Heritage*, which has less support for data curation and therefore suffers from typical issues such as ambiguous citations, lack of good quality metadata, lack of indexing in Crossref as well as in major indices. In this case, we expect both precision and contribution to OCC to be low.

Table 1. Evaluation outcomes: total references (*C-Refs*), overall matches (*M-Refs*), matches in the OCC (*O-Refs*), and OCC matches from Computer Science (*CS-Refs*).

Case	*C-Refs*	*M-Refs*	*O-Refs*	*CS-Refs*
Case 1	42	31 (73.8%)	9 (21.4%)	7
Case 2	30	15 (50.0%)	5 (16.5%)	4
Case 3	40	7 (17.5%)	n/a	n/a

For each paper (the citing entity), we measured the precision of the references returned by BCite by comparing the number of cited references matched by BCite (*M-Refs*) and the total number of references (*CRefs*) originally submitted by the user. So as to assess the contribution to the OpenCitations Corpus, we compared the number of cited references returned by BCite that have also a match in the OCC (*O-Refs*) with the total number of cited references returned by the process. For the sake of completeness, we also analyse how many of the references having a match in OCC do belong to the domain of Computer Science (*O-Refs*). All results are shown in Table 1.

Discussion. While the outcomes are at a very early stage, some conclusions can be already drawn. First, we can confirm the hypothesis that BCite matches references correctly even when lacking of data quality (Case 1), provided that citations are openly deposited (i.e. in the local BCite triplestore or in Crossref). Indeed, no mismatches were produced by BCite for the altered references but, for the three cases, the non-matched references were missing both in Crossref and in the OCC. Possible reason for this missings could be related with the publication date of the cited entities (older publications are less likely to be openly accessible and available in open indexes), non-English languages (e.g. publications in Italian) or specific domains (e.g. Humanities). This also suggests that BCite is a good support for editors to automatise the reconciliation of bibliographic data. Results also show that very few works have been found in OCC: we report approx. 15-20% of the matched references for the first 2 cases, and none for the third case. This is especially visible when compared to the ones gathered from Crossref ($|M\text{-}Ref|$ - $|O\text{-}Ref|$), which were approx. the double in all cases. This reveals that editors, in principle, could give a significant contribution to the OCC through addition of their data. In addition, a significant need for extending the coverage of the OCC also emerges from our results. This is also confirmed by the amount of matched references from the domain of Computer Science (*CS-Refs*), which are above 75% of the *O-Refs* for Case 1 and

Case 2. This can be interpreted as a current bias of the OCC towards Computer Science publications – which is reasonable, if we consider the fact that, currently, all the citing articles in the OCC comes from the Open Access subset of Europe PubMed Central, containing Life Science and Medical contributions mainly.

5 Related Work

Plenty of works have been published in the past years on the main topics touched in this paper: curation methodologies and tools, and the promotion and release of open scholarly data.

Curation methodologies have been largely studied over the years – see, for instance [1,5,7]. Tools supporting authors for data curation have been also proposed, such as Recite (`https://reciteworks.com/`) which however only supports in-text reference checking. Other tools have been released so as to help users and librarians to manage cross-citations (e.g. RefWorks, EndNote, Mendeley and Zotero, compared at `https://tinyurl.com/ap4k475.`), even if they do not offer the production of RDF data.

A number of initiatives for producing open scholarly data have been also proposed in the recent years, in addition to the OpenCitations Corpus [11]. For instance, OpenAIRE (`https://www.openaire.eu/`) provides a number of services (from deposition to discovery to statistics) for thousands of Open Access scholarly datasets, but currently does not include article citations. Springer Nature has recently published its SciGraph (`http://scigraph.springernature.com`), i.e. an open knowledge graph of Spinger Nature's scholarly data, even if it does not provide article citations yet. Scholarly Data (`http://www.scholarlydata.org/`) [8] allows the creation and exploration of data about Computer Science Conferences and Workshops. WikiCite includes a series of a collaborative activities to build a bibliographic database in Wikidata using a semi-automatic workflow. In the context of the Linked Open Citation Database, the work of [6] presents a semi-automated system for the creation and storage of citation data to support digital libraries in the data curation workflow.

6 Conclusions

In this paper we presented BCite, a service for bibliographic data curation and generation of RDF-based open citation data compliant with the OpenCitations Data Model. We showed how a simple tool based on a tit-for-tat approach can support journals (especially the smaller ones that have less resources for curatorial activities), aiding them in the process of bibliographic data cleaning, while also providing a contribution to an existing open citation data repository, i.e. the OpenCitations Corpus.

In the future, we plan to organise an extensive user-based evaluation of BCite in order to assess also its usability when it is used for addressing the task of bibliographic reference correction in real environments (e.g. in journals). In addition, we also plan to integrate BCite within the OpenCitations ingestion workflow, so as to directly

import additional open citation data into the OCC from the BCite App. In fact, the current implementation of BCite does not support such direct import, since a careful study about possible issues e.g. the license associated to the to-be-published articles used as input of the process, as well as the trustfulness of the possible BCite users and other data quality issues and compliancy with the evolving OpenCitations Data Model must be performed in advance with the OpenCitations team.

References

1. Bailey Jr, C.W.: Digital curation bibliography: Preservation and stewardship of scholarly works, 2012 supplement (2013), http://digital-scholarship.org/dcbw/
2. Biagetti, M.T.: An ontological model for the integration of cultural heritage information: Cidoc-crm. Italian Journal of Library, Archives and Information Science **7**(3), 43–77 (2016). https://doi.org/10.4403/jlis.it-11930
3. Citti, F.: Paolo bufalini and the classics: Towards a digital edition of his "note-book". Conservation Science in Cultural Heritage **8**(1), 65–89 (2008). https://doi.org/10.6092/issn.1973-9494/1396
4. Hammarfelt, B., Haddow, G.: Conflicting measures and values: How humanities scholars in australia and sweden use and react to bibliometric indicators. Journal of the Association for Information Science and Technology . https://doi.org/10.1002/asi.24043
5. Johnston, L.R.: Curating research data volume one: Practical strategies for your digital repository. Curating Research Data (2017), https://www.alastore.ala.org/content/curating-research-data-volume-one-practical-strategies-your-digital-repository
6. Lauscher, A., Eckert, K., Galke, L., Scherp, A., Rizvi, S.T.R., Ahmed, S., Dengel, A., Zumstein, P., Klein, A.: Linked open citation database: enabling libraries to contribute to an open and interconnected citation graph (2018)
7. Lord, P., Macdonald, A., Lyon, L., Giaretta, D.: From data deluge to data curation. In: Proceedings of the 3rd UK e-Science All Hands Meeting. pp. 371–375 (2004), http://www.ukoln.ac.uk/ukoln/staff/e.j.lyon/150.pdf
8. Nuzzolese, A.G., Gentile, A.L., Presutti, V., Gangemi, A.: Conference linked data: The scholarlydata project. In: In Proceedings of the 15th International Semantic Web Conference. pp. 150–158 (2016). https://doi.org/10.1007/978-3-319-46547-0_16
9. Peroni, S., Shotton, D.: Opencitations data model (2018). https://doi.org/10.6084/m9.figshare.3443876
10. Peroni, S., Shotton, D.: The spar ontologies. In: Proceedings of the 17th International Semantic Web Conference. Springer (2018)
11. Peroni, S., Shotton, D., Vitali, F.: One year of the opencitations corpus - releasing rdf-based scholarly citation data into the public domain. In: Proceedings of the 16th International Semantic Web Conference. pp. 184–192 (2017). https://doi.org/10.1007/978-3-319-68204-4_19
12. Renear, A.H.: Text encoding. A Companion to Digital Humanities **219**, 218–239 (2004). https://doi.org/10.1002/9780470999875.ch17