protocols.io

A methodology for gathering and annotating the raw-data/characteristics of the documents citing a retracted article

Dec 09, 2020

Ivan Heibi[1], Silvio Peroni[1]

[1]University of Bologna

1 *Works for me*   dx.doi.org/10.17504/protocols.io.bdc4i2yw

Ivan Heibi

ABSTRACT

Giving a retracted article, we present a step-by-step methodology for gathering the raw-data of the documents which have cited such article (starting from the date of its publication) and annotating the metadata and features for such citing entities. The external services used for our purpose are all free and open. The methodology uses three external services: (a) OpenCitations COCI (http://opencitations.net/index/coci, used to retrieve citation data), (b) RetractionWatch database (http://retractiondatabase.org, used to retrieve information of retracted articles), and (c) SCImago (https://www.scimagojr.com/, to retrieve subject areas and subject categories of publications). The methodology is divided into five steps: (1) identifying and retrieving the citing entities, (2) retrieving the citing entities characteristics, (3) classifying the citing entities according to subject areas and subject categories, (4) extracting textual values from the citing entities, and (5) annotating the in-text citations characteristics.

The application of this methodology produces a dataset containing all the citing entities and their annotated data/features. Starting from an empty dataset, each step of the methodology (from 1 to 5) enriches it with new variables.

EXTERNAL LINK

https://github.com/ivanhb/cits-ret-method

DOI

dx.doi.org/10.17504/protocols.io.bdc4i2yw

EXTERNAL LINK

https://github.com/ivanhb/cits-ret-method

PROTOCOL CITATION

Ivan Heibi, Silvio Peroni 2020. A methodology for gathering and annotating the raw-data/characteristics of the documents citing a retracted article. **protocols.io**
https://dx.doi.org/10.17504/protocols.io.bdc4i2yw

EXTERNAL LINK

https://github.com/ivanhb/cits-ret-method

KEYWORDS

citation analysis, citation function, retraction, methodology, python

CREATED

Mar 07, 2020

GUIDELINES

The aim of this methodology is to build a dataset containing the raw-data and the characteristics of the entities which have cited an examined retracted article. The methodology is divided into five steps which are summarized in the table below. For each step (table row) we mention its title (i.e. "Step" column), give a brief description (i.e. "Description" column), specify the inputs needed (i.e. "Input") and show the expected output (i.e. "Output" column). The output of each step associates new data/features to the gathered citing entities. Technically, it extends the final dataset with new variables (columns).

A	B	C	D
Step	**Description**	**Input**	**Output**
1) Identifying and retrieving the citing entities	Identifying the list of entities citing the retracted article and annotating their main metadata	DOI of the retracted article	*For each citing entity:* *1.1) DOI* *1.2) year of publication* *1.3) title* *1.4) venue id (ISSN/ISBN)* *1.5) venue title*
2) Retrieving the citing entities characteristics	Annotating whether the citing entities have been or have not been retracted as well	DOIs of the citing entities	*For each citing entity:* *2.1) is / is not retracted*
3) Classifying the citing entities according to subject areas and subject categories	Classifying the citing entities into areas of study and specific subject categories, following the SCImago classification	ISSN/ISBN of publication venues of the citing entities	*For each citing entity:* *3.1) subject area* *3.2) subject category*
4) Extracting textual values from the citing entities	Extracting the citing entities' abstracts, the in-text reference pointers, citation contexts, title of the section where the in-text citations happen	DOIs of the citing entities	*For each citing entity:* *4.1) abstract* *4.2) in-text citation section* *4.3) in-text citation context* *4.4) in-text reference pointer*
5) Annotating the in-text citations characteristics	Annotating the intent and sentiment of each in-text citation, and specifying whether the text in the citation contexts mentions the retraction of the cited article	In-text citations' contexts	*For each in-text citation:* *5.1) citation intent* *5.2) citation sentiment* *5.3) retraction is / is not mentioned*

An overview of all the steps needed for generating an annotated dataset for the citing entities of a retracted article. For each step, we give a brief description, the inputs needed, and the output. The output is represented as the expected list of annotated features that will enrich the final dataset.

All the related materials of this methodology are maintained in the dedicated Git repository at https://github.com/ivanhb/cits-ret-method. The Git repository contains the directories:

- "*data/*": a results sample of each step of the methodology
- "*img/*": the images used in this document
- "*doc/*": other documents related to the methodology
- "*script/*": the Python scripts and Python Notebooks

Some files/documents of the Git repository have been also included and uploaded into this document. The **method.py** script is the main script for launching the methodology. Each step is executed by running the main script with some dedicated parameters. The **method.py** script is also written in a Python Notebook format (.ipynb): **method.ipynb**.

ABSTRACT

Giving a retracted article, we present a step-by-step methodology for gathering the raw-data of the documents

which have cited such article (starting from the date of its publication) and annotating the metadata and features for such citing entities. The external services used for our purpose are all free and open. The methodology uses three external services: (a) OpenCitations COCI (http://opencitations.net/index/coci, used to retrieve citation data), (b) RetractionWatch database (http://retractiondatabase.org, used to retrieve information of retracted articles), and (c) SCImago (https://www.scimagojr.com/, to retrieve subject areas and subject categories of publications). The methodology is divided into five steps: (1) identifying and retrieving the citing entities, (2) retrieving the citing entities characteristics, (3) classifying the citing entities according to subject areas and subject categories, (4) extracting textual values from the citing entities, and (5) annotating the in-text citations characteristics.

The application of this methodology produces a dataset containing all the citing entities and their annotated data/features. Starting from an empty dataset, each step of the methodology (from 1 to 5) enriches it with new variables.

BEFORE STARTING

This methodology takes for granted some basic knowledge regarding the scholarly publishing nature, the usage of the references, and the in-text citation styles.

Before starting, you need to make sure you have Python3.x installed on your computer, in addition, in order to correctly execute the Python-based scripts indicated in the methodology, you must install the required libraries defined in **requirements.txt**. Please follow the official Python guidelines at https://wiki.python.org/moin/BeginnersGuide/ to check and eventually install python and the required libraries locally on your machine.

In the rest of this document we will use some common expressions/abbreviations, summarized in the following glossary:

- "*value*": the **values** are written in italic surrounded by quotation marks. In case the **value** itself contains quotation marks, then the quotation marks are written in italic style too: *"value"*.

- $Dataset$: a **dataset** represented in a tabular format. The first letter of the dataset name is in uppercase.

- M_{mem}^{D} : **M** is the **member** of the **dataset D** (a dataset can have only one member); The first letter of the member's name is written in uppercase.

- $\left[M_{mem}^{D}\right]$: a set containing the values of the **dataset (D) member (M)**.

- v_{var} : a **variable v.** The variable name is written in lowercase.

- $D[m, v_{var}]$: the **value** of the **variable v** for the corresponding **member m** of the dataset **D** (i.e. a table cell).

- $D[m, *]$: a set containing the **values** for a corresponding **member m** of the dataset **D** (i.e. a table row).

- $D[*, v_{var}]$: a set containing the **values** for a corresponding **variable v** of the dataset **D** (i.e. a table column).

$$D$$

M_{mem}^{D}	$v1_{var}$	$v2_{var}$	$v3_{var}$	$v4_{var}$
m0	... $D[m1,v2_{var}]$ ...		...	...
m1	...	a_value	...	...
m2	...	...	...	...
...	...	...	...	...

$[M_{mem}^{D}]$ $D[*, v3_{var}]$ $D[m2,*]$

Identifying and retrieving the citing entities

1 Starting from one retracted article identified with a DOI this step gets the metadata of all the citing entities included in the COCI dataset (the OpenCitations Index of Crossref open DOI-to-DOI references). We are only interested in a subset of attributes for the citing entities gathered. More specifically, for each citing entity we want to annotate:

- The DOI value
- The year of publication
- The title of the article
- The ID of the venue (ISSN/ISBN)
- The title of the venue

In practical terms, this step will initialize our main dataset and include the above attributes in it. The next steps of this methodology will further enrich the same dataset with new variables that characterize each citing entity of the dataset.

Input: DOI of the retracted article

Output: creates the dataset $Cits\_Dataset$ with the initial variables/columns: doi_{var}, $title_{var}$, $year_{var}$, $source\_id_{var}$, and $source\_title_{var}$.

doi	title	year	source_id	source_title
…	…	…	…	…

1.1 First, we need to set the retracted article we want to examine. We consider articles that have officially received one or more retraction notice and had been eventually fully retracted. The Retraction Watch service reports and collects information about the retractions of scientific papers which they make available in an open queryable database at http://retractiondatabase.org/. We use the Retraction Watch database to get the article we are interested in. Each record of the Retraction Watch database contains the following attributes (columns):

1. Title, Subject(s), Journal, Publisher, Affiliation(s), Retraction Watch Post URL(s)
2. Retraction reasons
3. Authors
4. The Original Paper date/PubMedID/DOI
5. The retraction notice date/PubMedID/DOI
6. Article type(s) and the nature of the notice
7. Countries, If it is Paywalled? and Other notes

For the proceeding of this methodology, we consider the following attributes from the above list:

1. The original DOI of the paper
2. The year of publication (although Retraction Watch reports the complete publication date we will only consider the year value)
3. The year of the retraction notice/s (some articles might have more than one retraction notice, we will consider all these notices).

Another aspect we need to take into consideration at this stage is the in-text citation style. We need to take note and keep in mind this information, which will become very important in the next steps.

1. The DOI of the retracted article
2. The year of publication
3. The year of the retraction notice/s

Example:
1. 10.1016/S0140-6736(97)11096-0
2. 1998
3. 2004, 2010

1.2 Now we need to get the list of the entities which have cited the retracted article. We will query the COCI dataset (https://opencitations.net/index/coci). This dataset contains details of all the citations that are specified by the open references to DOI-identified works present in Crossref (https://www.crossref.org/). OpenCitations provides a free APIs service to query and retrieve the COCI data at http://opencitations.net/index/coci/api/v1.

First, we get all the entities citing our retracted article using the "citations" operation: http://opencitations.net/index/coci/api/v1#/citations/. Once we have the list of all the citing entities, we outline each citing entity with the following attributes: (a) the DOI value, (b) the year of publication, (c) the title of the article: (d) the ID of the venue (ISSN/ISBN), and (e) the title of the venue. These attributes are available in the COCI datasets. We use the COCI APIs and apply the "metadata" operation: http://opencitations.net/index/coci/api/v1#/metadata/, which requests the DOI/s value of the entity we are looking for and returns the metadata of such entity (if any). The COCI API does not necessarily have the metadata of all our DOI values (citing entities), in this case, such citing entities are excluded from our analysis.

In this step, we initialize our main dataset and populate it with the citing entities and first metadata. For

the rest of this document, we will refer to our dataset as $Cits\_Dataset$. This step's

operations are done automatically by calling the following script.

```
Script to execute:
▪ python3 method.py -s 1.2 -in <DOI>

You can also specify a different output directory for the dataset:
▪ python3 method.py -s 1.2 -in <DOI> -out <DIR-PATH>

Example:
▪ python3 method.py -s 1.2 -in "10.1186/1756-8722-5-31"
▪ python3 method.py -s 1.2 -in "10.1186/1756-8722-5-31" -out path/to/dir
```

A dataset containing all the entities citing the retracted article

Example:

doi	title	year	source_id	source_title
10.1002/jppr1998286443	Controversies in Vaccination with Measles, Mumps, Rubella (MMR) Vaccine	1998	issn:2055-2335; issn: 1445-937X	Journal of Pharmacy Practice and Research
10.1177/1362361398024006	Clinical Features Associated with Autism: Observations of Symptoms Outside the Diagnostic Boundaries of Autistic Spectrum Disorders	1998	issn:1362-3613; issn:1461-7005	Autism
...	...	...	...	...

☐ **step_1_2.csv**

2   To give the citing entities other attributes that aren't part of the COCI metadata we need to use other services. The only thing we would like to check is whether any of the citing entities we are considering had been retracted as well. This value will be assigned to each citing entity of the $Cits\_Dataset$. At the end of this step, we will have an extended version of the $Cits\_Dataset$ which embeds the additional $is\_retracted_{var}$. The first substep prepares the $Cits\_Dataset$, while the second substep shows how to annotate the new variable.

Input: $Cits\_Dataset[*, doi_{var}]$

Output: extends the $Cits\_Dataset$ with the new variable $is\_retracted_{var}$

...	is_retracted
...	...

2.1   First, we need to prepare the $Cits\_Dataset$ for the upcoming annotation (done on the next substep). The dataset will be extended with the new variable, and its value set to "*todo*":
$Cits\_Dataset[*, is\_retracted_{var}]$ = "*todo*". This operation is done automatically by calling the following script.

```
Script to execute:
python3 method.py -s 2.1 -in "<Cits_Dataset-Path>"

Example:
python3 method.py -s 2.1 -in "output/cits_dataset.csv"
```

An extended version of the $Cits\_Dataset$

Example:

doi	title	year	source_id	source_title	is_retracted
10.1002/jppr1998286443	Controversies in Vaccination with Measles, Mumps, Rubella (MMR) Vaccine	1998	issn:2055-2335; issn: 1445-937X	Journal of Pharmacy Practice and Research	TODO
10.1177/1362361398024006	Clinical Features Associated with Autism: Observations of Symptoms Outside the Diagnostic Boundaries of Autistic Spectrum Disorders	1998	issn:1362-3613; issn:1461-7005	Autism	TODO
...	...	...	...	...	...

📊 **step_2_1.png**

2.2 To fill the new $is\_retracted_{var}$ we need to iterate over all the citing entities and manually verify whether any of the citing entities has been retracted as well. Again we use the RetractionWatch database (http://retractiondatabase.org/) and check all the citing entities using their DOI values. The "*todo*" values under $is\_retracted_{var}$ are substituted with a "*yes*"/"*no*" value depending on whether the examined DOI has/hasn't been fully retracted.

The $CitsDataset$ with the annotated $is\_retracted_{var}$

doi	title	year	source_id	source_title	is_retracted
10.1002/jppr1998286443	Controversies in Vaccination with Measles, Mumps, Rubella (MMR) Vaccine	1998	issn:2055-2335; issn: 1445-937X	Journal of Pharmacy Practice and Research	no
10.1177/1362361398024006	Clinical Features Associated with Autism: Observations of Symptoms Outside the Diagnostic Boundaries of Autistic Spectrum Disorders	1998	issn:1362-3613; issn:1461-7005	Autism	no
…	…	…	…	…	…

☐ **step_2-2.csv**

Classifying the citing entities according to subject areas and subject categories

3 The aim of this step is to annotate the subject area/s and subject category/s of each citing entity in $Cits\_Dataset$. To do this we consider the venue identifiers (ISSN/ISBN) and classify them into specific subject area/s and a subject category/s using the SCImago Journal Classification (https://www.scimagojr.com/). This classification groups the journals into a subject area (27 major thematics), and subject category (313 specific subject categories). These values define two different levels: (1) a macro layer for the subject area, and (2) a lower layer for a specific subject category.

In this step, we first focus on the citing entities having ISSN IDs, and then we move to analyze those having ISBN IDs. At the end of this step, the $Cits\_Dataset$ will be further extended with two additional variables: $area_{var}$ and $category_{var}$.

The first substep is a preparation phase. On substep 2 we handle the ISSN venues and on substep 3 and 4 we handle the ISBN venues. The final substep (i.e. 5) merges the results and populates the $Cits\_Dataset$.

Input: $Cits\_Dataset[*, source\_id_{var}]$

Output: extends the $CitsDataset$ with the new variables: $area_{var}$ and $category_{var}$

…	area	category
…	…	…

3.1 We first separate the ISSN and ISBN values into two datasets: $ISSN\_Dataset$ and $ISBN\_Dataset$. These datasets represent two indexes that include all the unique ISSN and ISBN values in the $Cits\_Dataset$. Both the datasets will have the $area_{var}$ and $category_{var}$ variables. The $ISBN\_Dataset$ contains the additional lcc_{var} variable (the reason will become clear on substep 3.3). The two indexes/datasets are generated automatically using the script below.

```
Script to execute:
python3 method.py -s 3.1 -in "<Cits_Dataset-Path>"

Example:
python3 method.py -s 3.1 -in "example_data/cits_dataset.csv"
```

$ISSN\_Dataset$: a dataset containing the unique ISSN IDs in the $Cits\_Dataset$

source_id	source_title	area	category
issn:0007-1048; issn:1365-2141	British Journal Of Haematology	TODO	TODO
issn:1356-1294	Journal Of Evaluation In Clinical Practice	TODO	TODO
…	…	…	…

step_3_1_1.csv

$ISBN\_Dataset$: a dataset containing the unique ISBN IDs in the $Cits\_Dataset$

source_id	source_title	lcc	area	category
isbn:9783319932231; isbn:9783319932248	Cognitive Errors And Diagnostic Mistakes	TODO	TODO	TODO
isbn:9783319932927; isbn:9783319932934	Autism In Translation	TODO	TODO	TODO
…	…		…	…

step_3_1_2.csv

3.2 We map each unique ISSN value of our index $ISSN\_Dataset$ into its corresponding area and category following the SCImago journal classification. This process is done manually by checking each ISSN value using the SCImago Journal Rank service at https://www.scimagojr.com/. Among the returned information and metadata, we have the subject area and subject category. Journals might have more than one subject area or subject category, we will take into consideration and write down all these values.

The following figure shows a result example from the Scimago Journal Rank service when searching for the ISSN value "*0273-9615*".

The subject area ($area_{var}$) and subject category ($category_{var}$) must be annotated inside the $ISSN\_Dataset$ following these rules:

- The "*;;*" segment (with white space at the end) is used as a separator between two different subject areas, and between two subject categories that belong to different areas.
- The "*;*" segment (with white space at the end) is used as a separator between two different subject categories that belong to the same area.

Consedining the above rules and the previous example (ISSN="*0273-9615*"), the correct form to annotate the $ISSN\_Dataset$ is:

- $area_{var}$: "*Medicine;; Psychology*"

- $category_{var}$: "*Pediatrics, Perinatology and Child Health;; Clinical Psychology; Developmental and Educational Psychology;;*"

An annotated $ISSN\_Dataset$ (the $area_{var}$ and $category_{var}$ are defined)

source_id	source_title	area	category
issn:0007-1048; issn:1365-2141	British Journal Of Haematology	Medicine	Hematology
issn:1356-1294	Journal Of Evaluation In Clinical Practice	Medicine	Health Policy; Public Health, Environmental and Occupational Health
...	...	...	...

☐ **step_3_2.csv**

3.3 We need to classify also the ISBN venues into their corresponding subject areas and subject categories. Again we use the Scimago Journal classification. This choice is based on the fact that our aim is to have a standard for all the venues regardless of their type (ISBN or ISSN).

The Scimago classification previously used for the ISSN sources belongs to the journal sources, therefore we can't apply a direct association of these values to the ISBN sources. We need a pre-elaboration which maps an ISBN classification model into the Scimago classification model (subject area and subject category).

The ISBN classification model we used is the Library of Congress Classification (LCC, https://www.loc.gov/catdir/cpso/lcco/). First, we need to assign for each ISBN source in the

$ISBN\_Dataset$ its corresponding LCC code. This operation is done manually using two

main services: (a) the ISBNDB service (https://isbndb.com/), and (b) Classify (http://classify.oclc.org/classify2/), an experimental classification web service.

The LCC code values are written under lcc_{var} in the $ISBN\_Dataset$.

An updated version of the $ISBN\_Dataset$ having lcc_{var} annotated

source_id	source_title	lcc	area	category
isbn:9783319625416; isbn:9783319625430	Representing Scientific Knowledge	Q375	TODO	TODO
isbn:9783319638225; isbn:9783319638232	Recurrent Respiratory Papillomatosis	RC168.P15	TODO	TODO

☐ **step_3_3.csv**

3.4 To compile the area and category of each ISBN source we call a function that maps the LCC codes to an area and category of the Scimago Journal classification. More precisely, this function will do the

following operations for each member m of the $ISBN\_Dataset$:

1. Considers only the starting alphabetic segment of $ISBN\_Dataset[m, lcc_{var}]$

   and find the corresponding LCC discipline using a pre-built lookup index. (e.g. "*RC360*" -> "*RC*" -> "*Medicine*")

2. Checks whether the value of the LCC subject is also a Scimago subject area using a pre-built Scimago index. If this is the case, the algorithm will automatically annotate the

$ISBN\_Dataset[m, area_{var}]$ with such value, and the

$ISBN\_Dataset[m, category_{var}]$ will have the same value with the addition of "*(miscellaneous)*" at the end of it, as it is done on the Scimago classification when denoting a Journal that treats general categories of a specific area. In case no corresponding Scimago area has been found the function moves to point 3.

3. Checks whether the value of the LCC subject is a Scimago subject category using the same pre-built Scimago index. If the corresponding value is present, the program will automatically annotate the

$ISBN\_Dataset[m, category_{var}]$ with such value, and the

$ISBN\_Dataset[m, area_{var}]$ will have the same value used on the Scimago classification to denote the macro area of such category. In case no corresponding Scimago category has been found the function moves to point 4.

4. The program will annotate both $ISBN\_Dataset[m, area_{var}]$ and

$ISBN\_Dataset[m, category_{var}]$ with the "*todo_manual*" value.

Once the above function completes its elaboration, we need to find the corresponding area and category for the records marked with the "*todo_manual*" value and annotate such values manually using the LCC index (http://www.loc.gov/catdir/cpso/lcco/). The above algorithm is executed by running the following script:

```
Script to execute:
python3 method.py -s 3.4 -in "<CitsDataset-Path>"

Example:
python3 method.py -s 3.4 -in "example_data/cits_dataset.csv"
```

An annotated $ISBN\_Dataset$ (the $area_{var}$ and $category_{var}$ are defined)

source_id	source_title	lcc	area	category
isbn:9783319625416; isbn:9783319625430	Representing Scientific Knowledge	Q375	Computer Science	Computer Science (miscellaneous)
isbn:9783319638225; isbn:9783319638232	Recurrent Respiratory Papillomatosis	RC168.P15	Medicine	Internal medicine

☐ **step_3_4.csv**

3.5 Finally, we merge the $ISSN\_Dataset$ and the $ISBN\_Dataset$ into our main $Cits\_Dataset$. The $Cits\_Dataset$ will be extended with the two variables $area_{var}$ and $category_{var}$.

Script to execute:
```
python3 method.py -s 3.5 -in "<CitsDataset-Path>"
```

Example:
```
python3 method.py -s 3.5 -in "example_data/cits_dataset.csv"
```

The $Cits\_Dataset$ with the annotated $area_{var}$ and $category_{var}$

doi	title	year	source_id	source_title	is_retracted	area	category
10.1002/jppr1998286443	Controversies in Vaccination with Measles, Mumps, Rubella (MMR) Vaccine	1998	issn:2055-2335; issn: 1445-937X	Journal of Pharmacy Practice and Research	no	Medicine	Pharmacology (medical);;
10.1177/1362361398024006	Clinical Features Associated with Autism: Observations of Symptoms Outside the Diagnostic Boundaries of Autistic Spectrum Disorders	1998	issn:1362-3613; issn:1461-7005	Autism	no	Psychology	Developmental and Educational Psychology
...	...	...	...	...	...	...	...

step_3_5.csv

Extracting textual values from the citing entities

4 In this step, we enrich the $Cits\_Dataset$ with new variables that denote some textual values contained in the citing entities' full-text. The values we are interested in are:

1. The abstract ($abstract_{var}$): the abstract of the citing entity (in case there is any).

2. The in-text citation context/s ($intext\_citation.context_{var}$): the textual context/s which contains a reference pointer of the retracted article

3. The in-text citation section/s ($intext\_citation.section_{var}$): the section/s which contains the reference pointer of the retracted article

4. The in-text citation pointer/s ($intext\_citation.pointer_{var}$): the in-text reference pointer (e.g. Heibi (2019))

The first substep prepares the $Cits\_Dataset$ to be filled later with the above values. Substep 4.2 discusses each one of the above values and indicates how to correctly annotate them.

Input: $Cits\_Dataset[*, doi_{var}]$

Output: extends the $Cits\_Dataset$ with the new variables: $abstract_{var}$, $intext\_citation.section_{var}$, $intext\_citation.context_{var}$ and $intext\_citation.pointer_{var}$.

…	abstract	intext_citation.section	intext_citation.context	intext_citation.pointer
…	…	…		…

4.1 We extend the $Cits\_Dataset$ with the new new variables: $abstract_{var}$, $intext\_citation.section_{var}$, $intext\_citation.context_{var}$ and $intext\_citation.pointer_{var}$. The default value assigned to these fields is "*todo*".

This process is made automatically by calling the following script:

```
Script to execute:
python3 method.py -s 4.1 -in "<Cits_Dataset-Path>"

Example:
python3 method.py -s 4.1 -in "example_data/cits_dataset.csv"
```

doi		abstract	intext_citation.section	intext_citation.context	intext_citation.pointer
10.1002/jppr1998286443	…	TODO	TODO	TODO	TODO
10.1177/1362361398024006	…	TODO	TODO	TODO	TODO
…		…	…	…	…

☐ **step_4_1.csv**

4.2 To annotate the new variables we need to examine the citing entities' full-text. Some full-texts are open and freely accessible, others are closed by paywalls. We consider only the entities that we can successfully access their full-text, all the others should be removed from the $Cits\_Dataset$ and not considered. Finding the full-texts and removing the citing entities (which lack a full-text) are operations to manually do (for each citing entity in the $Cits\_Dataset$).

Once we have collected all the full-texts, we need to replace the "*todo*" values with the true corresponding values following the rules below:

The abstract ($abstract_{var}$):

Copy the entire abstract from each citing entity's full-text. In case no abstract has been found, we write an empty string. Possible examples of documents lacking abstracts are book chapters or editorials.

The in-text citations pointer ($intext\_citation.pointer_{var}$):

To correctly annotate this variable we need to have a little background on the citing formats and how the reference pointers in the text are written. Look at the following guidelines: https://tinyurl.com/vtdd6x2 for a brief background on this topic.
We search inside the citing entities' full-text for all the in-text citation pointers of our retracted article, and we write down the value used to point to the retracted article reference entry. For instance, this means that for a member m in $Cits\_Dataset$, the value of

$$Cits\_Dataset[m, intext\_citation.pointer_{var}]$$ might be: *"Heibi(2019)".*

Note that this value is the same one adopted for each in-text citation inside the document, so

$$Cits\_Dataset[m, intext\_citation.pointer_{var}]$$ will have only one

value.

The in-text citations context ($intext\_citation.context_{var}$):

We want to write down the context of each detected in-text citation. We define our in-text citation context as the sentence that includes the in-text citation pointer (anchor sentence), plus the prior and the following sentence.
There are some special cases we need to handle. If the in-text citation pointer:
1. Appears in a title: the context equals the entire title.
2. Appears in a table cell: the context equals the entire table cell.
3. Appears in the first sentence of a section/sub-section: the context equals the anchor sentence plus the sentence after.
4. Appears in the last sentence of a section/sub-section: the context equals the anchor sentence plus the prior sentence.
We might have more than one in-text citation in one citing entity, in this case, we must include

the ";;" segment as a separator between every two different contexts. For instance, for a record m in

$$Cits\_Dataset$$, the value of

$$Cits\_Dataset[m, intext\_citation.context_{var}]$$ might be:

"*We will talk about this aspect later. As it was also observed in Heibi(2019). Now we move to the second point of our analysis. ;; This work takes into consideration new features. We are working on extending the previous work of Heibi(2019)*"

The in-text citations section ($intext\_citation.section_{var}$):

The section where the in-text citation appears. If the related citing entity's full-text does not include any section/paragraph (e.g. an editorial), then the value of

$$Cits\_Dataset[m, intext\_citation.context_{var}]$$ equals "*none*".

Otherwise, the in-text citation section is annotated using one/both these values:
1. **Type:** could be equal to one of the following values: (a) "*abstract*", (b) "*introduction*", (c) "*background*", (d) "*results*", (e) "*method*" and (f) "*conclusion*". We chose one of these values only if it is clearly inferred from the section title (e.g. the title contains the typology name). In case we can't link the section to any of these types we will take note of its position in the document: (a) "*first section*": appears in the first section of the article, (b) "*final section*": appears in the last section of the article, and (c) "*middle section*": it appears neither in the first section nor the final section.
2. **Title:** in case we have assigned to section one of the previous 5 typologies, we will omit this value. Otherwise, we will annotate the exact title surrounded by brackets (*"... "*). So the value of

$$Cits\_Dataset[m, intext\_citation.context_{var}]$$ will contain the title

alongside the section position annotated on point(1).

We consider only the sections on the first level, therefore the section of an examined in-text citation

should always refer to the higher one and not the inner subsections. For instance, if a citation occurs inside subsection 2.1 then the section to consider is 2.

We might have more than only one in-text section in each examined citing entity, in this case, we must include the "*;;*" segment as a separator between two different sections, and we use the "*,*" as a separator between the type and title of the section.

For instance, this means that for a record m in $Cits\_Dataset$, the value of

$$Cits\_Dataset[m, intext\_citation.section_{var}]$$ in case it contains two

in-text citations is:
"*introduction;; final section; "Discussion" *"

step_4_2.csv

5   In this step, we add to the $Cits\_Dataset$ three variables that characterise the annotated in-text citation/s:

1. $intext\_citation.intent_{var}$: the citation intent/reason/function: the author's reason for citing a specific paper (e.g. because the citing document want to use the method defined in the cited paper),

2. $intext\_citation.sentiment_{var}$: the author's sentiment regarding the cited entity. We check whether the author's sentiment toward the cited entity is positive/negative/neutral.

3. $intext\_citation.ret\_mention_{var}$: check whether at least one of the in-text citations (of the examined citing entity) does explicitly mention the fact that the cited entity is retracted.

In the first substep, we will prepare the $Cits\_Dataset$, while the next substeps discuss how to correctly annotate it.

Input: $Cits\_Dataset[*, intext\_citation.context_{var}]$

Output: extends the $Cits\_Dataset$ with the new variables:

$$intext\_citation.intent_{var}\,,\ intext\_citation.sentiment_{var}\,,\ \text{and}$$

$$intext\_citation.ret\_mention_{var}\,.$$

...	intext_citation .intent	intext_citation .sentiment	intext_citation .ret_mention
...	...	...	...

5.1

This substep extends the $Cits\_Dataset$ with the new variables:

$$intext\_citation.intent_{var}\,,\ intext\_citation.sentiment_{var}\,,$$

and $intext\_citation.ret\_mention_{var}$. The default value assigned is "*todo*".

This process is made automatically by calling the following script:

```
Script to execute:
python3 method.py -s 5.1 -in "<Cits_Dataset-Path>"

Example:
python3 method.py -s 5.1 -in "example_data/cits_dataset.csv"
```

doi		intext_citation.intent	intext_citation.sentiment	intext_citation.ret_mention
10.1002/jppr1998286443	...	TODO	TODO	TODO
10.1177/1362361398024006	...	TODO	TODO	TODO
...		...	...	...

step_5_1.csv

5.2

All the variables of this step are manually inferred from the intent citation context. To correctly replace the default initial "*todo*" values we should follow the rules below.

The in-text citation intent ($intext\_citation.intent_{var}$ **):**

This variable answers the question "Why the citing entity is citing the retracted article?", so we want to

12/09/2020

Citation: Ivan Heibi, Silvio Peroni (12/09/2020). A methodology for gathering and annotating the raw-data/characteristics of the documents citing a retracted article. https://dx.doi.org/10.17504/protocols.io.bdc4i2yw

This is an open access protocol distributed under the terms of the **Creative Commons Attribution License** (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

examine the intent/reason of the citation. The CiTO ontology, the Citation Typing Ontology (https://sparontologies.github.io/cito), is an ontology for the characterization of factual and rhetorical bibliographic citations. Although the CiTO ontology characterizes also the in-text citations lacking an explicit in-text citation pointer, we will not consider these variants. Instead, we perform the analysis on the in-text citations previously annotated (on Step-4) which appear in the full-text with an in-text citation pointer.

On CiTO the citation intents are the object properties ($CiTO^{op}$), the

$intext\_citation.intent_{var}$ is compiled using one of $CiTO^{op}$ values. Despite

the fact that an in-text citation might refer to more than only one $CiTO^{op}$, our work restricts the

decision to only one value. This decision should simplify the future elaborations on the annotated dataset and limit the possible ambiguities.

We have designed a CiTO decision-model, to help us decide which $CiTO^{op}$ value assign for an

examined in-text citation in case we have more than one suitable $CiTo^{op}$ value. This model is

based on a priority ranked strategy. The following figure shows a graphical representation of the model.

Considering a member m of $Cits\_Dataset$, our decision-model works as follow:

1. We read the $Cits\_Dataset[m, intext\_citation.context_{var}]$, and

   find the most suitable citation intent for it. The above model presents 3 macroblocks, we outline the suitable one/s considering the analyzed in-text citation context. We can take a cue from the description, the usage, and the example of each block. Notice that the analyzed in-text citation context might be suitable for more than only one block.

2. Once we have chosen a suitable macroblock/s, we move toward a deepen selection of the suitable citation intent/s (CiTO object property). At the end of this phase, we will have a set of citation

   intents (based on CiTO): $Intent\_set = \{x : x \in CiTo^{op}\}$.

3. In case we have chosen only one value: "x", then the value of

   $Cits\_Dataset[m, intext\_citation.intent_{var}]$ is "x". Otherwise (the

   $Intent\_set$ contains more than one $CiTO^{op}$) we take a decision based on a priority

   approach as described in the next point.

4. To calculate the priority of a $CiTO^{op}$ value: "x", we sum the corresponding y-axis and x-axis

   values, along with its cell inner value. The smaller a value, the more priority it has. For instance,

$$priority(\text{``}confirms\text{''}) = 11.2 \text{ is higher than}$$

$$priority(\text{``}describes\text{''}) = 43.2 \text{ . We will calculate the priority of each value "}x\text{"}$$

in the $Intent\_set$, and select the one with the higher priority value.

Since a record m in $Cits\_Dataset$ might have N in-text citations, we must include the ",;"

segment as a separator between two different $intext\_citation.intent_{var}$ values.

The in-text citation sentiment ($intext\_citation.sentiment_{var}$ **):**

After reading each $intext\_citation.context_{var}$ we will annotate the

$intext\_citation.sentiment_{var}$ with one of the following values:

- "*positive*": the retracted article was cited as a valid prior work, and its findings/conclusions could have been also used in the citing study.
- "*negative*": the citing study cites the retracted article but addresses its findings as inappropriate/valid.
- "*neutral*": the author cites the retracted article without including any judgment or personal opinion regarding its validity.

We must include the ",;" segment as a separator between two different

$intext\_citation.sentiment_{var}$ values.

The in-text citation mentions the retraction (

$intext\_citation.ret\_mention_{var}$ **):**

We look at the value of $intext\_citation.context_{var}$ and check whether at least one

of the in-text citation contexts explicitly mentions the fact that the cited entity is retracted. Notice that here we are not interested in characterizing the in-text citation. We rather want to annotate

$intext\_citation.ret\_mention_{var}$ with a singular value: "*yes*"/"*no*" .

To make this annotation as much as possible coherent and less subject to language ambiguities, we decided to annotate with a "*yes*" value, only in case the word "retract" and its derivatives are explicitly used when addressing the cited entity in at least one of the in-text citations contexts in

$intext\_citation.context_{var}$.

doi		intext_citation.intent	intext_citation.sentiment	intext_citation.ret_mention
10.1002/jppr1998286443	…	disputes;; discusses;; disputes	neutral;; neutral;; neutral	no
10.1177/1362361398024006	…	discusses;; cites as evidence	neutral;; neutral	no
…		…	…	…

step_5_2.csv