# Data Citation: a Computational Challenge

## [Extended Abstract]

Susan B. Davidson
University of Pennsylvania
Philadelphia, PA
susan@cis.upenn.edu

Peter Buneman
University of Edinburgh
Edinburgh, UK
opb@inf.ed.ac.uk

Daniel Deutch
Tel Aviv University
Tel Aviv, Israel
danielde@post.tau.ac.il

Tova Milo
Tel Aviv University
Tel Aviv, Israel
milo@post.tau.ac.il

Gianmaria Silvello
University of Padua
Padua, Italy
silvello@dei.unipd.it

## ABSTRACT

Data citation is an interesting computational challenge, whose solution draws on several well-studied problems in database theory: query answering using views, and provenance. We describe the problem, suggest an approach to its solution, and highlight several open research problems, both practical and theoretical.

## Keywords

Data citation; Data provenance; Query answering using views

## 1. THE PROBLEM

Citation is essential to traditional scholarship. It helps identify the cited material so that it can be retrieved, gives credit to the creator of the material, dates it, and so on. In the context of printed materials, such as books and journals, citation is well understood. However, the world is now digital, and an increasing amount of information is being collected in structured and evolving curated databases, driving database owners, publishers and standards groups to consider how such data should be cited.

Currently, several databases describe (in English) what "snippets" of information are to be included in a citation for information displayed as web page views of the database. Three such examples are eagle-i[1], an RDF dataset built to facilitate translational science research which allows researchers to share information about resources such as cell lines and software; Reactome[2], an open-source, curated and peer reviewed pathway relational database; and Drugbank[3], a relational database combining chemical, pharmacological and pharmaceutical data with sequence, structure, and pathway information. Instructions are given on which snippets of information on the web page view of the resource should be included in a citation to the resource. However, generating the citation is left to the user.

Another interesting example is the IUPHAR/BPS Guide to Pharmacology [4] (GtoPdb). The content of GtoPdb represents the effort of a large number of members of the scientific community. Different portions of the database, with varying granularity, are contributed and/or curated by different subgroups of these individuals. While GtoPdb as a whole can be treated as a traditional publication and cited accordingly, citations to many web page views of the database include the people who provided the relevant content, along with an identifier for the view and the version of the database. Views are generated on the fly from the database, and citations are generated at the same time.[5] Thus, GtoPdb automatically generates citations, but only for *some* queries.

The question then is: How can we generate citations for *general queries* over the database, i.e. those which do not correspond to web page views of the database? The problem is that, unlike traditional publications which have a fixed granularity to which citations can be attached – e.g. a conference proceedings, or a paper in a conference proceedings – the *granularity* of the material to be cited varies. Since there are a potentially infinite number of queries, each accessing and generating different subsets of the database, we cannot hope to explicitly attach a citation to every possible result set and/or query. Instead, we must find ways of using citations for some portions of the database to automatically construct citations for more general queries. Thus data citation is a *computational* problem, as argued in [4].

The approach we propose next draws inspiration from results in two areas that have been extensively studied by the database theory community: *query answering using views* (e.g. [9, 3, 10]) and *database provenance* (e.g. [8, 5]). In particular, we leverage the fact that citations and provenance are both forms of *annotation* that are manipulated through queries. We give an overview of the approach next; details can be found in [6].

[1]https://www.eagle-i.net/
[2]http://www.reactome.org/
[3]http://www.drugbank.ca/

[4]http://www.guidetopharmacology.org/
[5]Since versioning is not enabled, re-executing the query brings back the current version which may be different from the version seen when cited.



*PODS'17 May 14-19, 2017, Chicago, IL, USA*



ACM ISBN 978-1-4503-4198-1/17/05.

DOI: http://dx.doi.org/10.1145/3034786.3056123

## 2. AN APPROACH

We start with a set of *view queries* that are specified by the database owner, representing the web-page views or portions of the database for which citations are understood. Each view query $V$ is associated with one or more *citation queries* and a *citation function*, which are also specified by the database owner. The citation queries pulls snippets of information from the database to be included in the citation; the citation function takes the output of the citation queries as input and outputs a citation in some appropriate format (e.g. human readable, BibTex, RIS or XML). The citation becomes an *annotation* on every tuple $t$ in the view.

View and citation queries are expressed as Conjunctive Queries [2] that are optionally *parameterized* by one or more variables. The parameters must appear in the head of the queries, and be consistent across the view and associated citation queries. In a parameterized view, subsets of tuples which agree on all parameter values will have the same citation, which is based on the output of the citation query when its parameters are instantiated with these values. Consequently, tuples which disagree on some parameter value may have different citations. We denote by $C_{\mathtt{V}}(p_1, ..., p_n)$ the citation for the tuples in the result of a view $V$ parameterized with $p_1, ..., p_n$.

For example, suppose we have the following relations where the keys are underlined.[6] For each `Family` tuple, the `Committee` relation associates the names of committee members responsible for the content (`Desc`) of a particular family. The `FamilyIntro` relation contains additional information (`Text`) about a particular family, which may be contributed by people other than committee members (details omitted):

```
Family(FID, FName,Desc)
Committee(FID, PName)
FamilyIntro((FID, Text)
```

The following view query is parameterized by `FID`, and therefore creates a separate citation for each tuple in `Family`. The associated citation query pulls the names of committee members from the `Committee` relation for the family tuple, in addition to its `FID` and `FName` from the `Family` relation:

```
λ FID. V1(FID,FName,Desc):- Family(FID,FName,Desc)
λ FID.CV1(FID,FName,PName):- Family(FID,FName,Desc),
                             Committee(FID,PName)
```

However, in the following (unparameterized) view all tuples in `Family` have the same citation, indicated by the associated citation query:

```
V2(FID,FName,Desc):- Family(FID,FName,Desc)
CV2(D):- D="IUPHAR/BPS Guide to PHARMACOLOGY..."
```

Our approach to constructing the citation to a general query is to *rewrite* it to a set of equivalent queries using the views, and combine the citations for these views to construct a citation to the general query. To do this, we leverage the fact that citations and provenance are both forms of *annotation* that are manipulated through queries [5]. In particular, the *joint* ($\cdot$) and *alternative* ($+$) use of annotations within a rewriting are modeled using the *semirings* approach of [8].

Given a query $Q$ and set of views $\mathcal{V}$ together with their associated citations, consider the set of minimal equivalent rewritings, $\{Q_1, ..., Q_n\}$. In the rewritings, parameters are ignored.

For example, consider the query

$$\mathtt{Q}(\mathtt{FName}) : -\mathtt{Family}(\mathtt{FID}, \mathtt{FName}, \mathtt{Desc}), \mathtt{FamilyIntro}(\mathtt{FID}, \mathtt{Text})$$

and an additional (unparameterized) view `V3` and associated citation query:

```
V3(FID,Text):- FamilyIntro(FID,Text)
CV3(D):- D="IUPHAR/BPS Guide to PHARMACOLOGY..."
```

`Q` can be rewritten in terms of `V1` and `V3`, or `V2` and `V3`:

```
Q1(FName):- V1(FID,FName,Desc),V3(FID,Text)
Q2(FName):- V2(FID,FName,Desc),V3(FID,Text)
```

To define the citation for a tuple in the result of the query, we start by defining a citation for a *single binding* of a *single rewriting* of the query. This dictates a single output tuple, and a particular valuation to the parameters of the views. We define the citation of the output tuple as the joint use of citations for the views and the parameter valuations, denoted by "$\cdot$".

DEFINITION 2.1. *Let $Q$ be a query and let $V$ be a set of citation views, such that $C_{\mathtt{Vi}}$, $F_{Vi}$ are respectively a citation query and citation function associated with $Vi$. Further let $Q'$ be a (partial) rewriting of $Q$ using $V1, ..., Vn \in V$. Further let $B$ be a binding to the variables of $Q'$, yielding an output tuple $t$. The citation for $t$ w.r.t. $Q, Q', V, B$, denoted $cite(t, Q, Q', V, B)$, is defined as*

$$cite(t,Q,Q',V,B) = F_{V1}(C_{\mathtt{V1}}(B1)) \cdot \ldots \cdot F_{Vn}(C_{\mathtt{Vn}}(Bn))$$

*where $Bi$ is the result of applying $B$ to the variables occurring in an atom involving $Vi$ in $Q'$.*

Multiple bindings lead to multiple *alternative* citations, which we capture using $+$.

DEFINITION 2.2. *Let $Q, V, Q'$ be as in Definition 2.1, and let $\beta_t$ be the set of all bindings for $Q'$ that yield a tuple $t$. The citation for $t$ w.r.t. $Q, Q'$ (denoted $cite(t, Q, Q', V)$) is defined as*

$$cite(t, Q, Q', V) = \Sigma_{B \in \beta_t} cite(t, Q, Q', V, B).$$

As noted earlier, a query may have multiple rewritings, each leading to a possibly different citation for a tuple. These are again alternatives, but the function used to combine the citations for them may be different than the one used for multiple bindings for a single rewriting. We therefore use $+^R$ ("$+$ for rewritings") to denote this function. Note that this is a formal semantics, not a means of computation: going through all rewritings would be impractical.

Returning to our example, suppose that there are two families that share the name 'Calcitonin' and therefore two sets of bindings for the result tuple (Calcitonin):

```
{FID=11, FName='Calcitonin', Desc='C1', Text='1st'}
{FID=12, FName='Calcitonin', Desc='C2', Text='2nd'}
```

Using rewriting `Q1`, the citation for the result tuple 'Calcitonin' would be $C_{\mathtt{V1}}(\mathtt{11}) \cdot C_{\mathtt{V3}} + C_{\mathtt{V1}}(\mathtt{12}) \cdot C_{\mathtt{V3}}$, where 11 and 12 are the parameters passed to `V1`. Using rewriting `Q2`, the citation would be $C_{\mathtt{V2}} \cdot C_{\mathtt{V3}}$.

The resulting citation for result tuple 'Calcitonin' would therefore be:

[6]The example is drawn from GtoPdb, and `Family` refers to families of drug targets.

$$(C_{\texttt{V1}}(\texttt{11})\cdot C_{\texttt{V3}} + C_{\texttt{V1}}(\texttt{12})\cdot C_{\texttt{V3}}) +^R (C_{\texttt{V2}}\cdot C_{\texttt{V3}})$$

Finally, to obtain a citation for the query answer, the citations for its member tuples must be combined. We denote this abstract function as *Agg*.

The abstract functions "$\cdot$" , "+", $+^R$ and *Agg* are *policies* to be specified by the database owner. There are many interpretations that could be used for these functions. For "$\cdot$", "+" and *Agg*, union or join are natural. For $+^R$, the "minimum" in some ordering would also be natural. The ordering could reflect how "precise" or "comprehensive" the rewritings are relative to each other, or the "size" of the resulting citation.

As a final step in our example, suppose we use union for "$\cdot$", "+" and *Agg*, and $+^R$ as the (estimated) minimum size. Since `V1` is parameterized, $C_{\texttt{V1}}$ will be different for every tuple. In contrast, $C_{\texttt{V2}}$ and $C_{\texttt{V3}}$ will be the same for every tuple. The estimated size of the citation using `Q1` would therefore be proportional to the size of `Family`, whereas the estimated size of the citation using `Q2` would be 1 (recall that *Agg* is union). The final citation for `Q` would therefore be the citation leading to the minimum size, which is the one using `Q2` ($C_{\texttt{V2}}\cdot C_{\texttt{V3}}$).

# 3. DISCUSSION AND OPEN PROBLEMS

A number of interesting research questions arise from this approach, both practical and theoretical.

**Calculating citations.** While we have defined a model for citations for query results, we have not given an efficient means for computing them. In particular, it is infeasible both in terms of run time and the size of the resulting citation to go through all rewritings and all assignments within each of them, pointing to the need for cost functions to reduce the search space (somewhat analogous to e.g. [9, 3, 10]). It may also be possible to do some of the reasoning at the schema level, and impose the views that are retained at this level over tuple-level annotations.

**Defining citations.** As described, views represent the web-page views or common queries against the database for which citations are understood. From a practical perspective, the database owner must first ensure that the database includes the snippets of information to be included in the citation queries. They must then specify the view and citation queries in terms of the database schema, as well as the policies for combining citations ("$\cdot$", "+" and *Agg*, and $+^R$). This could easily be overwhelming for a non-expert, and therefore designing a user-friendly interface with appropriate defaults is essential. Note that once the view and citation queries have been specified, the system should take care of the annotation tracking rather than relying on the database owner to modify the database schema.

From a theoretical perspective, there are interesting questions around defining and efficiently deciding whether these views represent the "best" ones given an expected query workload, i.e. the ones that "cover" the expected queries, and give concise and unambiguous results.

**Fixity.** One of the "core principles" of data citation [1, 7] is *fixity*: data may evolve over time, and a citation should bring back the data as seen at the time it was cited. Thus the citation must include a mechanism of obtaining the data, in addition to snippets of information useful for human understanding. One approach is for the database to support versioning, and citations to include a timestamp or version number along with the query (or some means of recovering the query) used to obtain the data. (A prototype system for relational databases which pursues these ideas can be found in [11].)

**Size of citations.** An obvious concern is that, since views may be parameterized, the size of a citation may be proportional to the size of the query result. In contrast, conventional citations are small – for example, when there is an extended author list (more than 3 authors), we use "et al" to abbreviate – and much of this is driven by wanting a reasonable size for the bibliography section of conventional publications. How does this change when we are talking about digital objects? In particular, should the citation object returned be an encoding of or reference to an extended citation which is a searchable object?

**Citation evolution.** The views or the citations associated with views may change over time, either in response to a change in query workload or evolving standards in data citation. This can be captured in our model by including a "timestamp" attribute in base relations, with lambda variables in views corresponding to this attribute. Citations could then depend on the timestamp. An intriguing computational challenge is how to compute citations in an incremental manner in this setting.

**Other models.** Conjunctive queries are a core for many different models and languages, and the idea of citation views is useful beyond relational systems, e.g. XML and RDF. However, do we need to go beyond conjunctive queries? In particular, for several of the RDF systems we have examined the citation depends on the *class* of resource and determining the class of the resource involves reasoning over an ontology. What extensions to the language are needed, or are there other approaches that might be useful for RDF systems?

# 4. ACKNOWLEDGMENTS

This work has been partially funded by NSF IIS 1302212, NSF ACI 1547360, and NIH 3-U01-EB-020954-02S1; by the European Research Council under the FP7, ERC grant MoDaS, agreement 291071; by the Israeli Science Foundation (1636/13); and by a grant from the Blavatnik Interdisciplinary Cyber Research Center.

# 5. REFERENCES

[1] *Out of Cite, Out of Mind: The Current State of Practice, Polocy, and Technology for the Citation of Data*, volume 12. CODATA-ICSTI Task Group on Data Citation Standards and Practices, September 2013.

[2] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[3] F. N. Afrati, C. Li, and J. D. Ullman. Using views to generate efficient evaluation plans for queries. *J. Comput. Syst. Sci.*, 73(5):703–724, 2007.

[4] P. Buneman, S. Davidson, and J. Frew. Why data citation is a computational problem. *CACM*, 59, 2016.

[5] J. Cheney, L. Chiticariu, and W. C. Tan. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–474, 2009.

[6] S. B. Davidson, D. Deutch, T. Milo, and G. Silvello. A model for fine-grained data citation. In *CIDR 2017,*

*8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*, 2017.

[7] FORCE-11. *Data Citation Synthesis Group: Joint Declaration of Data Citation Principles*. FORCE11, San Diego, CA, USA, 2014.

[8] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS*, pages 31–40, 2007.

[9] A. Y. Halevy. Answering queries using views: A survey. *VLDB J.*, 10(4):270–294, 2001.

[10] L. Popa and V. Tannen. An equational chase for path-conjunctive queries, constraints, and views. In *Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel, January 10-12, 1999, Proceedings.*, pages 39–57, 1999.

[11] S. Pröll and A. Rauber. Scalable data citation in dynamic, large databases: Model and reference implementation. In *Proc. of the 2013 IEEE International Conference on Big Data*, pages 307–312, 2013.