By Dongwon Lee, Jaewoo Kang, Prasenjit Mitra, C. Lee Giles, and Byung-Won On

ARE YOUR CITATIONS CLEAN?

If they are, only one can refer to a distinct document; if not, many can refer to the same document.

"Citations" play an important role in many scientific-publication digital libraries (DLs), including CiteSeer, arXiv e-Print, DBLP, and Google Scholar. By "citation," we mean the collection of bibliographic information (such as author name, article title, publication venue, and year published) pertinent to a particular article. Users often use citations to find information of interest in DLs, and researchers depend on citations to determine the impact of a particular article in a DL. In addition, when DLs are integrated, citations serve as unique identifiers of associated documents. Therefore, citations of stored documents in DLs must be consistent and up-to-date. But maintaining consistency is generally nontrivial. Challenges include: data-entry errors, citation formats, lack of (enforcement of) standards, imperfect citation-gathering software, common author names or abbreviations of publication venue, and large-scale citation data.

Many of these problems can be solved through "global IDs," no matter how different two citations might seem; if both carry the same global ID, they are considered to be the same citation. Popular global IDs include ISBNs and digital object identifiers (DOIs) [10]. Despite their many benefits, however, publishers have only partially adopted them, and users have largely ignored them (especially on the Web). That is, scholars who post their "publication list" to their home pages usually do not put a DOI ahead of each citation. Similarly, they usually do not use DOIs in the reference section when writing scientific documents, though there are exceptions among scientific disciplines (such as physics). Even if all such users would adopt global IDs, interoperation among different global IDs (such as ISBN vs. DOI) would still be an issue. Moreover, marking existing documents with global IDs is costly. For DLs whose data is manually curated by human experts (such as Thomson Scientific's Science Citation Index and DBLP), the issue of erroneous and duplicate citations is less obvious but still exists. However, for DLs in which data is gathered and generated automatically by software (such as CiteSeer and Google Scholar), the problem is exacerbated [8]. Since automated indexing methods [4] are not as accurate as human experts, and human users use diverse citation formats to refer to the same article, many citation errors are included in these DLs. For large-scale DLs in which human indexing methods are not sustainable, it is essential for DL administrators to employ highly accurate automated methods.

As a result, in order to maintain clean citations, DLs must routinely search their collections and fix incorrect citations or remove duplicates. This so-called citation matching (CM) problem is a special version of a more general problem known as the "record linkage" problem [3, 12], which has been researched in various disciplines under a variety of names [2, 6, 9, 11]. The CM problem can be stated as: Given two lists of citations, A and B, for each citation a in A, find a set of citations b in B such that both a and b refer to the same article.

In practice, to determine whether or not two citations refer to the same real-world document (without using global IDs), people use some distance metrics (such as Levenstein, Jaro, and Cosine) and a predefined similarity threshold. That is, according to some distance function, if the distance between two citations, a and b, is within the threshold, then two citations are marked as "duplicates."

Figure 1. Screenshot of citation search for "Russell and Norvig" in Google Scholar; the result includes redundant citations, all referring to the same book.

To demonstrate the need for a solution to the CM problem, we present three problems drawn from real applications. The first is an example introduced in [8]. Figure 1 is a screenshot of Google Scholar in which a user is searching for a book *Artificial Intelligence: A Modern Approach* by S. Russell and P. Norvig. Note that Google Scholar keeps multiple citations (with different formats) of the same book, mistakenly implying that each is different. However, all 23 actually refer to the same book published by the same authors and thus should have been consolidated in the DL.

The second problematic example is drawn from the ACM Portal[1], which contains the names of all the authors who have ever published in the ACM DL. However, as shown in Figure 2, the name of the

[1]Since 2005, when we first reported the CM problem to ACM, the ACM Portal team has been working on a massive author name normalization project to resolve it.

author "Jeffrey D. Ullman" is spelled in a variety of ways, including eight variants under "Ullman" and two under "Ullmann." As a consequence, Ullman's citations are divided and mislabeled into 10 different duplicate author entries. Such errors often indirectly contribute to the CM problem. The third example is an inverse case of the second example. It is drawn from DBLP, a popular computer science DL, where users are able to browse a collection of articles grouped by the author's full name or where an author's full name acts as a primary key. Figure 3 is a screenshot of a collection of articles by "Wei Wang," but there are at least four prolific computer scientists with the same name "Wei Wang" spelled the same way. Not surprisingly, their citations are all mixed here. Any bibliometric analysis using this data would be faulty.

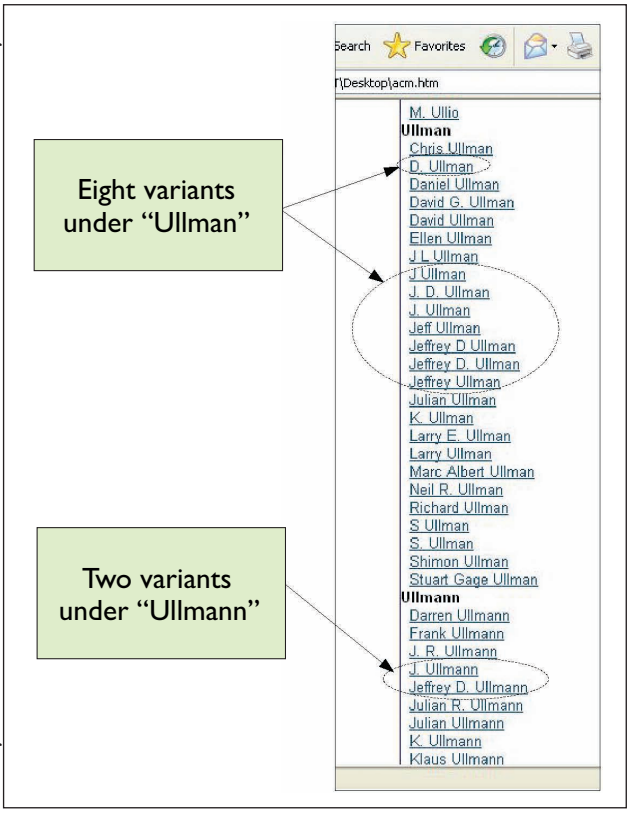

Figure 2. Screenshot of author index for "Ull*" in the ACM Portal; the citations for "Jeffrey D. Ullman" appear as eight variants under "Ullman" and two variants under "Ullmann."

In general, since different users use different citation formats, DLs may contain a variety of citations, all referring to the same document. Automatically determining (and eliminating) duplicates in such a DL is not only nontrivial it may be impossible. Nonetheless, the CM problem in DLs is important and must be solved. If we could identify and match citations precisely, we would enable precise bibliometric analyses. This would result in attributing credit to the correct authors, identifying all citations to a given article, and analyzing the effect of scholarly articles more accurately.

Moreover, the CM problem arises not only in the context of DLs but in many other related contexts. For example, online product catalog services (such as Google's Froogle) face similar problems. They extract product descriptions (such as product name, price, and manufacturer) from different Web pages and consolidate the extracted information into lists such that all information related to the same product goes into the same list. This problem is (in a broad sense) a CM problem. Different Web pages use different conventions to represent the same information. Solutions that address a CM problem should be applicable to this problem as well.

Scenarios

We refer to a set of citations (or a DL) in which the CM problem has not been solved as "dirty"; otherwise, we refer to a DL as "clean." That is, in a clean DL, there is at most one citation that refers to a distinct article in the real world, while in a dirty DL, more than one citation referring to the same real-world document may exist; for instance, CiteSeer today is a "dirty" DL. So far, the CM problem has been considered in a rather narrow sense, but the DLs of the new generation face new scenarios (see Table 1):

Creation. When a new DL is created from a collection of digital literature, the citation entries are typically extracted first from the literature; the extracted citation entries are then cleaned and matched. In order to handle a large number of citations, CM in this scenario generally involves two steps:

Blocking. The citation entries are grouped into blocks based on some inexpensive distance metrics or by sorting on some key values (such as title or author's last name); and

Matching. The algorithms visit each block separately and perform more elaborate matching within a block.

Most previous work on the CM problem sought to address this scenario. Formally, Given a set of dirty citation entries, S, find all clusters C ($\subset S$), such that all entries in C are close to one another with respect to some distance function.

Insertion. Once a DL is created, it needs to be kept up-to-date by adding new articles and their citations over time. Unlike the creation scenario, insertion occurs almost daily throughout its lifetime. For instance, CiteSeer crawls the Web searching for new literature, indexing them as new documents are found. In this scenario, the set of newly found citations is inserted into an already established clean DL

(where all duplicates are consolidated). Although the CM problem in this scenario occurs frequently, it is largely ignored by both the CM and the record-linkage communities. Efficient handling of insertion is important for maintaining a large-scale DL. Formally, Given a set of dirty citation entries S_a (that are newly found) and a set of clean citation entries S_b (that is, the existing DL), for each entry a ($\in S_a$), find a closest entry b ($\in S_b$), such that *dist(a,b)* is less than or equal to *t,* where *dist* is some distance function and *t* is a threshold.

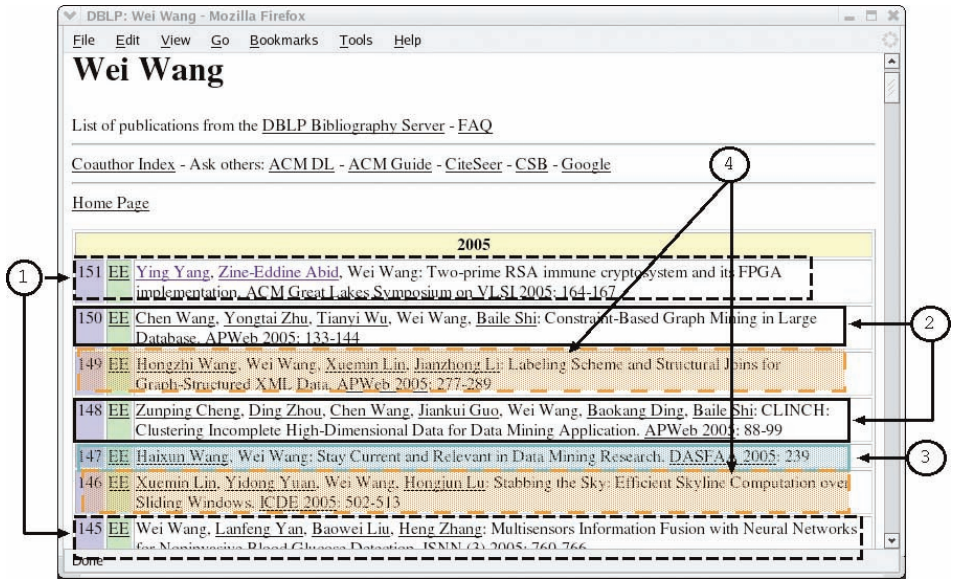

Figure 3. Screenshot of a collection of citations under the author "Wei Wang" in DBLP; at least four distinct computer scientists share the name "Wei Wang."

Integration. This scenario occurs when merging multiple DLs (such as CiteSeer and arXiv). The basic assumption is that in each DL, citation entries are already cleaned, and in most cases duplicates are eliminated (by possibly going through the previous creation and insertion scenarios). Therefore, CM mainly concerns the problem of linking citation entries across the DLs that refer to the same object. As in the insertion scenario, to the best of our knowledge, little CM work has been done in the integration context. Formally, Given two sets of clean citation entries, S_a and S_b, find a one-to-one mapping between entries, a ($\in S_a$) and b ($\in S_b$), such that *dist(a,b)* is less than or equal to *t.*

Scenario	S_a	S_b	Characteristics
Creation	Dirty	-	-
Insertion	Dirty	Clean	$S_a \neq S_b$ and $\lvert S_a \rvert << \lvert S_b \rvert$
Integration and Interoperation	Clean	Clean	$S_a \neq S_b$

Table 1. Four scenarios for creating and maintaining DLs.

Interoperation. In response to a query over a federated system of DLs, CM must be performed on the intermediate results obtained from the individual DLs before they are returned to the end user. As in the integration scenario, the citations (referring to the same real-world article but presumably obtained from different DLs and potentially having different formats) must be matched. As in the integration scenario, we assume that the DLs themselves are clean and that duplicates have been eliminated; still, duplicates in the intermediate results from different DLs must be removed. Formally, Let S_a and S_b be the sets of clean citation entries in the results returned from two different DLs in response to a federated search. Find a one-to-one mapping between entries, a ($\in S_a$) and b ($\in S_b$), such that *dist(a,b)* is less than or equal to *t.* As seen in the similarities in the definitions of the interoperation and integration scenarios, the same CM algorithms can be used for both.

Challenges

Although the CM problem (and its general version, the record-linkage problem) has been studied extensively in many disciplines, including databases, statistics, DLs, and artificial intelligence, existing techniques are unable to cope with the new DL challenges, as discussed in the following paragraphs:

CM solutions focus mainly on the creation scenario. But as DLs proliferate, their usage patterns and working scenarios change as well. For instance, the federation of multiple DLs using the standards developed by the Open Archive Initiative (www.openarchives.org) is no longer a distant dream. Also the characteristics of each scenario are slightly different; thus an efficient solution for one scenario does not necessarily work well for another scenario. Therefore, the ability to handle the insertion and merge scenarios is crucial in the new generation of DLs.

We've seen a dramatic increase in recent years of both the number of DLs available on the Web and the volume of data maintained in all DLs. For instance, about 356 known DLs had been developed through the National Science Foundation's National Science Digital Library program (as of 2004). Furthermore, some existing DLs include a large number of citations (tens of millions) (see Table 2). However, most CM solutions focus on a rather static collection of small to mid-size DLs (including from 1,000 to 10,000 citations) [2, 7, 8, 11]. According to esti-

Since different users use different citation formats, DLs may contain a variety of citations, all referring to the same document.

mates, CiteSeer indexes 10 million citation records [4]. Detecting and reconciling variants among 10 million citations efficiently is not a trivial task without compromising accuracy (recall the problem in Figure 1). The accuracy of existing CM solutions leaves much room for improvement.

Although several previous studies have reported an impressive 80%–95% CM accuracy in their experiments [2, 8, 11], their applicability is limited when related methods are applied to large-scale DLs. Note that a plain nested-loop-based CM algorithm requires all pair-wise comparisons of citations—a quadratic time complexity. Since it is computationally expensive for a large data set, typical CM algorithms involve a preprocessing stage called "blocking" to select a smaller candidate set for further examination. Although blocking schemes vary, it is not uncommon for thousands of citations in the candidate set to require further examination after blocking. Therefore, when such CM methods must be "repeatedly" applied to "very large" citation data, performance is still important.

Digital Library	Domain	# of Citations (in Millions)	Automatically Constructed?
ISI/SCI	General Science	25	No
CAS	Chemistry	23	No
MEDLINE/PubMed	Life Science	12	No
CiteSeer	General Science, Engineering	10	Yes
arXiv e-Print	Physics, Mathematics	0.3	No
SPIRES HEP	High-Energy Physics	0.5	No
DBLP	Computer Science	0.6	No
CSB	Computer Science	1.4	Yes
NetBib	Network	0.05	No

Table 2. Characteristics of several well-known scientific DLs.

In light of today's generation of supercomputers (more than 10 teraflops of processing power) this computation may appear to be achievable. However, the citations typically reside on disk. Though disk speeds have increased, quadratic computation over very large data sets is still not feasible. Besides, DLs may be unable (for financial reasons) to employ more powerful supercomputers to perform these computations. Furthermore, in light of the quality-of-service implications, the administrators of merging DLs may not want these computations run over the DLs for long periods. Therefore, developing novel solutions capable of achieving scalability and accuracy remains a challenge.

Nonstandard formats. Despite recent efforts to standardize citation formats (such as the Open Citation Project, opcit.eprints.org), authors of articles collected in DLs will continue to use nonstandard formats. Due to the lack of enforcement mechanisms, formats vary by personal taste, journal policy, and discipline. For instance, citations in some engineering fields require at least author name and paper title, while those in physics may not require a paper title. Citations in the engineering and physical sciences may use unique identifiers for citations, while those in the social sciences may lack identifiers. Similarly, a recommended citation format in one journal is likely to be quite different from the citation format in another. Citation formats posted on the Web are even more diverse.

Therefore, DLs with citations collected from the Web tend to suffer from more serious ambiguity. Consider the citations in Figure 1. Although they can all refer to the same book, and minor problems like variations in spacing, line breaks, or hyphenation can be resolved through simple rules, it is difficult to resolve problems resulting from different citation formats. The differences occur in citation formatting, including: number of fields used, order of fields, field values, typos or personal comments, special characters like space or hyphen, and XML. Developing solutions capable of handling a variety of formats is a challenge.

Public access. Although the record-linkage industry continues to grow, few CM systems (or even record-linkage systems) are available to the research community; examples include Carnegie Mellon University's SecondString, GNU EPrints, and ParaTools. A system needs to be developed and made available for easy access by the public.

Conclusion

Despite its importance and potential benefit to the DL community, the CM problem is seriously underresearched. Due to its unique aspects (such as large number of available fields), generic solutions developed for the record-linkage problem do not necessarily work that well. Furthermore, the novel challenges faced by today's DLs cannot be handled easily through existing solutions. To emphasize the importance of the problem, we've presented a preliminary "rethinking" of new challenges important in contemporary DLs.

References

1. Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval.* Addison-Wesley, 1999.
2. Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., and Fienberg, S. Adaptive name-matching in information integration. *IEEE Intelligent Systems 18,* 5 (Sept./Oct. 2003), 16–23.
3. Fellegi, I. and Sunter, A. A theory for record linkage. *Journal of the American Statistical Society 64* (1969), 1183–1210.
4. Giles, C., Bollacker, K., and Lawrence, S. CiteSeer: An automatic citation indexing system. In *Proceedings of the ACM Conference on Digital Libraries* (Pittsburgh, PA, 1998), 89–98.
5. Hong, Y., On, B.-W., and Lee, D. System support for name authority control problem in digital libraries: OpenDBLP approach. In *Proceedings of the European Conference on Digital Libraries* (Bath, U.K., Sept. 2004), 134–144.
6. Jaro, M. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association 84,* 406 (June 1989), 414–420.
7. Jin, L., Li, C., and Mehrotra, S. Efficient record linkage in large data sets. In *Proceedings of the International Conference on Database Systems for Advanced Applications* (Kyoto, Japan, Mar. 2003), 137–148.
8. Lawrence, S., Giles, C., and Bollacker, K. Digital libraries and autonomous citation indexing. *IEEE Computer 32,* 6 (June 1999), 67–71.
9. On, B.-W., Lee, D., Kang, J., and Mitra, P. Comparative study of the name disambiguation problem using a scalable blocking-based framework. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries* (Denver, CO, 2005), 344–353.
10. Paskin, N. DOI: A 2003 progress report. *D-Lib Magazine 9,* 6 (June 2003).
11. Pasula, H., Marthi, B., Milch, B., Russell, S., and Shpitser, I. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems.* MIT Press, Cambridge, MA, 2003.
12. Winkler, W. *The State of Record Linkage and Current Research Problems.* Tech. Rep., U.S. Bureau of the Census, Washington, D.C., Apr. 1999; www.census.gov/srd/papers/pdf/rr99-04.pdf.

Dongwon Lee (dongwon@psu.edu) is an assistant professor in the College of Information Sciences and Technology at the Pennsylvania State University, University Park, PA.

Jaewoo Kang (kangj@korea.ac.kr) is an assistant professor in the Department of Computer Science and Engineering at Korea University, Seoul, Korea.

Prasenjit Mitra (pmitra@ist.psu.edu) is an assistant professor in the College of Information Sciences and Technology at the Pennsylvania State University, University Park, PA.

C. Lee Giles (giles@ist.psu.edu) is the David Reese Professor in the College of Information Sciences and Technology at the Pennsylvania State University, University Park, PA.

Byung-Won On (on@cse.psu.edu) is a Ph.D. candidate in the Department of Computer Science and Engineering at the Pennsylvania State University, University Park, PA.