Maintaining an Online Bibliographical Database: The Problem of Data Quality

Michael Ley*, Patrick Reuther*

*Department for Databases and Information Systems, University of Trier, Germany
{ley,reuther}@uni-trier.de
http://dbis.uni-trier.de http://dblp.uni-trier.de

Abstract. CiteSeer and Google-Scholar are huge digital libraries which provide access to (computer-)science publications. Both collections are operated like specialized search engines, they crawl the web with little human intervention and analyse the documents to classify them and to extract some metadata from the full texts. On the other hand there are traditional bibliographic data bases like INSPEC for engineering and PubMed for medicine. For the field of computer science the DBLP service evolved from a small specialized bibliography to a digital library covering most subfields of computer science. The collections of the second group are maintained with massive human effort. On the long term this investment is only justified if data quality of the manually maintained collections remains much higher than that of the search engine style collections. In this paper we discuss management and algorithmic issues of data quality. We focus on the special problem of person names.

1 Introduction

In most scientific fields the amount of publications is growing exponentially. The primary purpose of scientific publications is to document and communicate new insights and new results. On the personal level publishing is a sort of collecting credit points for the CV. On the institutional level there is an increasing demand to evaluate scientists and departments by bibliometric measures, which hopefully consider the quality of the work. All aspects require reliable collection, organization and access to publications. In the age of paper this infrastructure was provided by publishers and libraries. The internet, however, enabled new players to offer services. Consequently many specialized internet portals became important for scientific communities. Search engines like Google(-Scholar) or CiteSeer, centralized archives like arXic.org/CoRR and a huge number of personal and/or department web servers make it very easy to communicate scientific material.

The old players — publishers, learned societies, libraries, database producers etc. — face these new competitors by building large digital libraries like ScienceDirect (Elsevier), SpringerLink, ACM Digital Library or Xplore (IEEE) in the field of computer science.

DBLP (*Digital Bibliography & Library Project*) (Ley, 2002) is an internet "newcomer" that started service in 1993. The DBLP service evolved from a small bibliography specialized to *database systems* and *logic programming* to a digital library covering most subfields

of computer science. Today (October 2005) DBLP indexes more than 675.000 publications published by more than 400.000 authors and is accessed more than two million times a month on the main site maintained at our department.

To build a bibliographic database always requires decisions between quality and quantity. You may describe each publication by a very rich set of metadata and include classifications, citation links, abstracts etc. — or restrict it to the minimum: authors, title, and publication venue (journal, book, Web-address). For DBLP we decided for the minimalistic approach, our very limited resources prevent us to produce detailed metadata of a substantial number. For each attribute of the metadata the degree of consistency makes the difference: It is easy to produce a huge number of bibliographic records without standardization of journal names, conference names, person names, etc. As soon as you try to guarantee that an entity (journal, conference, person, ...) is always represented by exactly the same character string and no entities share the same representation, data maintenance becomes very expensive. Traditionally this process is called *authority control*. In DBLP the number of different journals is a few hundreds, the number of different conference series a few thousands. To guarantee consistency on this scale requires some care, but is is not a real problem. Even for a moderate sized bibliographic database like DBLP, authority control for person names is much harder: the magnitude is $> 400K$ and the available information often is incomplete and contradictory.

2 Process Driven Data Quality Management

Data Quality comprises many different dimensions and aspects. Redman presents a variety of dimension such as the completeness, accuracy, correctness, currency and consistency of data, just to name a few (Redman, 1996). Other aspects are the unambiguity, credibility, timeliness, meaningfulness. A good overview on different dimensions of data quality can be obtained from (Dasu and Johnson, 2003) (Scannapieco et al., 2005).

Information acquisition is a critical phase for data quality management. For DBLP there is a broad range of primary information sources. Usually we get electronic documents, but sometimes all information have to be typed in. Some important sources like SpringerLink for the *Lecture Notes in Computer Science* series provide basic information in a highly structured format which is easy to transform into our internal formats. For many very diversely formatted sources it is not economic to develop wrapper programs, we have to use a standard text editor and/or adhoc scripts to transform the input to a format suitable for our software.

In some cases we have only the front pages (title pages, table of contents) of a journal or proceedings volume. The table of contents often contains information inferior to the head of the article itself: Sometimes the given names of the authors are abbreviated. The affiliation information for authors often is missing. Many tables of contents contain errors, especially if they were produced under time pressure like many proceedings. Even in the head of the article itself you may find typographic errors.

A very simple but important policy is to enter all articles of a proceedings volume or journal issue in one step. In DBLP we make only very few exception from this *all or nothing policy.* For data quality this has several advantages over entering CVs of scientists or reference lists of papers: It is more easy to guarantee complete coverage of a journal or conference series. There is less danger to become biased in favor of some person(s). Timeliness is only to achieve, if new journal issues or proceedings are completely entered as soon as they are published.

A very early design decision was to generate *author pages*: For each person who has (co)authored (or edited) a publication indexed in DBLP our software generates an HTML-page which enumerates her/his publications and provides hyperlinks to the coauthors and to the tables of contents pages the article appeared in. From the database point of view these are simple materialized views, for the users they make it very convenient to browse in the person–person and person–publication graphs. The graph implied by the coauthor relationship is an instance of a social network (Watts, 2004) (Staab, 2005), the DBLP coauthor net was recently used to analyse the structure of several sub-communities of computer science (Hassan and Holt, 2004) (Elmacioglu and Lee, 2005) (Liu et al., 2005).

We interpret a new publication as a set of new edges in the coauthor graph — or as an incrementation of the weights of existing edges. For each new publication we try to find all authors in the existing collection. We use several simple search tools with a variety of matching algorithms, in most cases traditional regular expressions are more useful than any tricky distance functions. The lookup is essentially a manual process driven by intuition and experience how to find most efficiently person names which might be misspelled or incomplete. Usually this manual lookup process is organized in two levels: We hire students to do the formatting (if necessary) and a first lookup pass. They annotate articles or names which require further investigation or more background knowledge. Often the students find incomplete or errorless entries in the database. In a second pass over the table of contents the problematic cases are treated and errors in the database are corrected (this is done by M. Ley). Finally the new information is entered into the database. During this stage a lot of simple formatting conventions are checked by scripts, for example we are warned if there are consecutive upper case characters in a person name.

At a typical working day we add $\sim$500 bibliographic records to DBLP. It is unrealistic to belief that this is possible without introducing new errors and without overlooking old ones. It is unavoidable that care during the input process varies. The obvious dream is to have a tool which does the hard work — or more realistic — which helps us to do it. To approach this goal we tried to understand how we find errors and inconsistencies most efficiently.

Often it is very helpful to look at the neighbourhood of a person in the coauthor graph. Because most scientific publications are produced by groups, many errors show up locally. A first important milestone to facilitate manual inspection was the development of the DBL-Browser (Klink et al., 2004) as a part of the SemiPort project (Fankhauser et al., 2005). The DBL-Browser provides a visual user interface in the spirit of Microsoft Explorer: A mixture of tree visualizations with folder and document icons and web-style hypertext make it very easy to navigate inside and between author pages. For persons with long publication lists the chronological listing provided by our web interface becomes insufficient, selections by coauthor, journal/conference, year, etc. are very helpful. The main-memory DB underlying the DBL-Browser guarantees short latency times. This is a very important factor for the usability of the system: fast reaction makes it practical "to snoop around" and to find suspicious entries.

The process driven strategy which tries to avert errors from getting into the database by controlling and improving the information acquisition process should be complemented by a more data driven strategy which tries to detect and correct errors in the existing data (Redman, 1996).

3 Data Driven Quality Management

Data driven strategies can be divided into *database bashing* and *data edits*. The key idea behind database bashing is to compare or crosscheck the data stored in one database to different data sources such as an other database or information from real world persons in order to find errors or to confirm the quality of the original data. Database bashing is useful for error detection, however the correction of the errors is troublesome. If there are differences between two records - that are assumed as records describing the same entity from different sources - the question arises which of the two records is correct, or whether any of those to occurrences is hundred percent correct. Data edits do not focus on the comparison of records from different sources but make use of business rules. These business rules are specific to the domain of the database. For the domain of bibliographic records such a rule is for example: "Alert us if there are authors in the dataset that slightly vary in their spelling but have exactly the same co-authors".

Exactly this rule was implemented by a simple software: We build a data structure which represents the coauthor graph. Our algorithm checks all pairs (a_1, a_2) of author nodes which have the distance 2 in the graph. If the names of these nodes are very similar, we should suspect them to represent the same person:

$$\textbf{if } StringDistance(name(a_1), name(a_2)) < t \textbf{ then warning}$$

The *StringDistance* function and the threshold value t required some experimentation. At the moment a modified version of the classical Levenshtein distance is in use, it implements special rules for diacritical characters (umlauts, accents, etc.) and for abbreviated name parts. The program produces a list of several thousand warnings. The main problem are not the false drops, but the suspicious pairs which can not be resolved because of lack of information. In many cases we are able to find the missing part of the puzzle — for example on personal "home pages" of the scientists themselves, but often the information is not available with a reasonable effort. We soon found that it is more economical to look only at persons whose publication lists has been modified recently. For these persons it is more likely to resolve contradictory spellings or to complete abbreviated name parts.

The simple software sketched above is in daily use since 2 years. It helped us to locate a large number of errors, but it should be replaced by an improved system for two reasons: We still find too much errors more or less accidentally and not by a well understood search process. The precision of the warnings still is too low — we spend too much time on suspicious pairs of names we can not resolve. Because the time we can invest for error corrections is very limited, we need a tool which points us to the most promising cases.

4 A Framework for Person Name Matching / Outlook

We are now experimenting with a much more flexible software framework for person name matching. The key ideas are:

- It makes no sense to apply distance functions to all pairs of person names in our collection because this product space is too large ($O(n^2)$ algorithms for $n > 400000$) and because comparing totally unrelated names produces too many false drops. Our *distance*

2 in the coauthor graph heuristic is a (quite successful) example of a *blocking function*. A *block* is a set of person names which are somewhere "related", *blocking* is defined as a set of blocks. A person name may be a member of several blocks inside a blocking. Distance functions are applied only to all tuples drawn from a block and not from the much larger set of all names — the complexity now is dominated by the size of the largest block. A blocking function is an algorithm which produces a blocking.

- A very rich set of *distance functions* is described in the literature. An excellent starting point to explore them is the SecondString project (Bilenko et al., 2003). Our software makes it easy to plug in new distance functions and to combine them. For person names very domain specific functions seem to be useful, for example to match transcriptions of chinese names.
- The system is implemented as a *data streaming architecture* very similar to a query processor in a data base management system. This well understood architecture gives much flexibility to add operators like union, intersection, materialization, loading of older results, selection, etc.

The starting point for the new software was the Java-based reimplementation of our well-tried algorithms within the new framework. The next step was the addition of several distance functions and stream operators. To make the resulting lists of warnings more useful, each block inside a blocking got a label — for example the name of the person building the connection between the two suspects for the distance 2 blocking, or the name of the conference/journal both have published in, or the title word both used in some of their publications. This annotation is propagated through the stream. A typical output of our system looks like this:

Brian T. Bennett(2) – (Peter A. Franaszek) & (journals/ibmrd) – Brian T. Bennet(2)

There are 2 occurrences of the name *Brian T. Bennett* and 2 others with a single 't'. They share the coauthor *Peter A. Franaszek* and both have published in the IBM Journal of Research and Development.

For software the open source movement produced a fascinating variety of systems which often are competitive to commercial software. For the narrow field of metadata for computer science publications such a "open database" culture is nearly missing. The only exception is the sharing of BibTeX files in *The Collection of Computer Science Bibliographies* founded by Christian Achilles. Since a few years the DBLP data set is available in XML (*http://dblp.uni-trier.de/xml*). To our surprise this had a very interesting impact: (1) We are aware of > 100 publications which use the DBLP data as a test data set for a very broad range of experiments, most in the field of XML processing. (2) Several groups published papers about the name disambiguation problem and used the DBLP data as a main example (Lee et al., 2004) (On et al., 2005) (Han et al., 2005) .... Our next steps will be to understand the details of these articles, to reimplement the proposed methods within our framework, and to test them in our daily work.

Acknowledgements: The most encouraging kind of quality control is user feedback. We appreciate all e-mails by users, we hope that no serious mails become victims of our rigid spam filters. We try to correct all errors we are pointed to immediately. Unfortunately it is far beyond our resources to include all publications we are asked to consider. At the moment DBLP is supported by the Microsoft Bay Area Research Center and by the Max-Planck-Institut für Informatik. We hope to find more sponsors ...

References

Bilenko, M., R. J. Mooney, W. W. Cohen, P. Ravikumar, and S. E. Fienberg (2003). Adaptive name matching in information integration. *IEEE Intell. Syst. 18*(5), 16–23.

Dasu, T. and T. Johnson (2003). *Exploratory Data Mining and Data Cleaning*. John Wiley.

Elmacioglu, E. and D. Lee (2005). On six degrees of separation in DBLP-DB and more. *SIGMOD Record 34*(2), 33–40.

Fankhauser, P. et al. (2005). Fachinformationssystem Informatik (FIS-I) und semantische Technologien für Informationsportale (SemIPort). In *Informatik 2005, Bd. 2*, pp. 698–712.

Han, H., W. Xu, H. Zha, and C. L. Giles (2005). A hierarchical naive bayes mixture model for name disambiguation in author citations. In *SAC 2005*, pp. 1065–1069. ACM.

Hassan, A. E. and R. C. Holt (2004). The small world of software reverse engineering. In *WCRE*, pp. 278–283.

Klink, S., M. Ley, E. Rabbidge, P. Reuther, B. Walter, and A. Weber (2004). Browsing and visualizing digital bibliographic data. In *VisSym 2004*, pp. 237–242.

Lee, M.-L., W. Hsu, and V. Kothari (2004). Cleaning the spurious links in data. *IEEE Intelligent Systems 19*(2), 28–33.

Ley, M. (2002). The dblp computer science bibliography: Evolution, research issues, perspectives. In *SPIRE 2002, Lisbon, Portugal, September 11-13, 2002*, pp. 1–10. Springer.

Liu, X., J. Bollen, M. L. Nelson, and H. V. de Sompel (2005). Co-authorship networks in the digital library research community. *CoRR cs.DL/0502056*.

On, B.-W., D. Lee, J. Kang, and P. Mitra (2005). Comparative study of name disambiguation problem using a scalable blocking-based framework. In *JCDL 2005*, pp. 344–353.

Redman, T. C. (1996). *Data Quality for the Information Age*. Artech House.

Scannapieco, M., P. Missier, and C. Batini (2005). Data quality at a glance. *Datenbank-Spektrum 14*, 6–14.

Staab, S. (2005). Social networks applied. *IEEE Intell. Syst. 20*(1), 80–93.

Watts, D. J. (2004). *Six Degrees: The Science of a Connected Age*. NY: W. W. Norton.

Résumé

CiteSeer et Google Scholar sont des bibliothèques électroniques gigantesques donnant accès à des publications scientifiques (en informatique). Ces deux collections sont gérées comme des machines de recherche spécialisées qui parcourent le web avec peu d'interventions humaines et analysent les documents pour les classer et pour extraire des méta-données des textes complets. D'autre part, il y a aussi des bases de données bibliographiques comme INSPEC en ingéniérie et PuBMed en médecine. En informatique, le service DBLP a évolué d'une petite bibliographie en une bibliothèque électronique couvrant la plupart des domaines de l'informatique. Les collections du second groupe sont gérées avec un effort humain considérable. À long terme, un tel investissement n'est justifié que si la qualité des données reste très supérieure a celle de collections de type machines de recherche. Dans cet article, nous discutons les aspects gestion et algorithmique de la qualité des données. Nous nous concentrons sur le problème particulier des noms de personnes.