Record Matching in Digital Library Metadata

Min-Yen Kan and **Yee Fan Tan**
Department of Computer Science
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
{kanmy, tanyeefa}@comp.nus.edu.sg

When data stores grow large, data quality, cleaning and integrity become issues. The commercial sector spends a massive amount of time and energy canonicalizing customer and product records, as their lists of products and consumers expand. An Accenture study in 2006 found that a high-tech equipment manufacturer saved US$6 million per year by removing redundant customer records used in customer mailings. In 2000, the UK Ministry of Defence embarked on the massive "The Cleansing Project", solving key problems with their inventory and logistics, and saving over US$25 million over four years.

In digital libraries, such de-duplication problems manifest most urgently not in the customer, product or item records, but rather in the metadata that describe the library's holdings. Several well-known citation lists of computer science research contain over 50% duplicate citations, although none of these duplicates are exact string matches [2]. Without metadata cleaning, libraries may end up listing multiple records for the same item, causing circulation problems and skewing the distribution of its holdings. In addition, when different authors share the same name (e.g., Wei Wang, J. Brown), author disambiguation must be performed to correctly link authors to their respective monographs and articles, and not to others. Metadata inconsistencies can be due to problems with varying ordering of fields, type of delimiters used, omission of fields, multiple representations of names of people and organizations, and the occasional typographical error.

When libraries import large volumes of metadata from sources that follow a metadata standard, a manually compiled set of rules called a crosswalk may used to transform the metadata to the library's own format. However, such crosswalks are expensive to create manually and public ones exist only for a few, well-used formats. Crucially, they also do not address how to detect and remove inexact duplicates. As digital libraries mine and incorporate data from a wider variety of sources, especially noisy sources such as the Web, finding a suitable and scalable matching solution becomes a pressing need.

Let us examine this problem and its solutions. The de-duplication task takes a list of metadata records as input and returns the list with duplicate

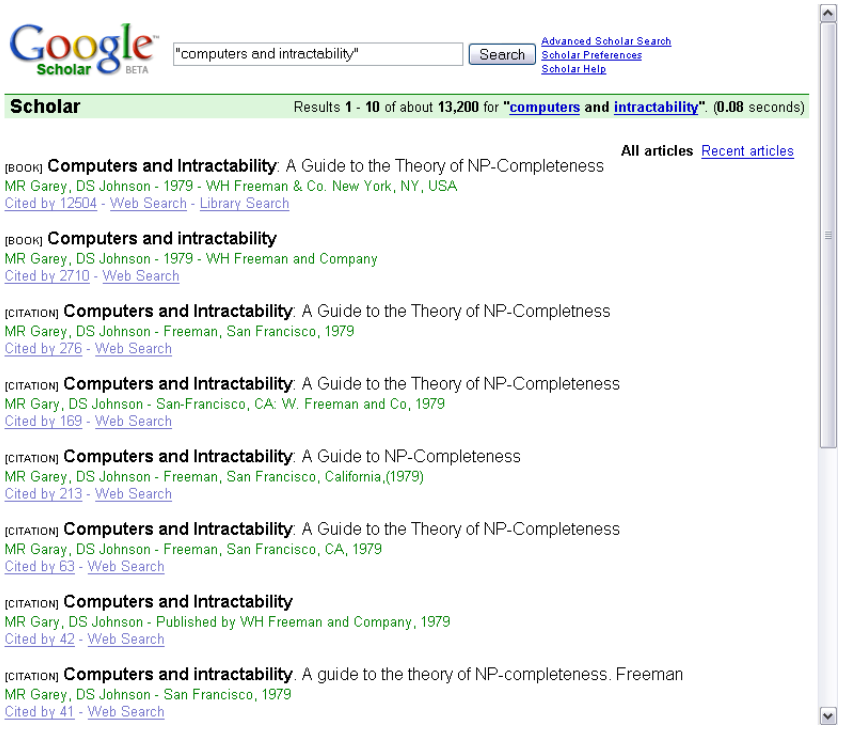

Figure 1: Searching for "computers and intractability" on Google Scholar.

records removed. For example, the search results shown in Figure 1 are identical and should have been combined into a single entry. It should be noted that many disciplines of computer science have instances of similar inexact matching problems, and as such this problem has many names including de-duplication, data cleaning, disambiguation, record linkage, attribution, entity resolution and plagiarism detection. While these variant problems differ in specifics, a common key operation is to determine whether two data records match. A key goal of this article is to introduce this problem and generate awareness of the approaches espoused by different communities. For a detailed review, we urge readers to consult the individual papers and a more detailed survey paper [5].

1 Uninformed String Matching

In its most basic form, record matching can be simplified as string matching, which decides whether a pair of observed strings refer to the same underlying item. In such cases, we use the similarity between the strings to calculate whether they are coreferential. When such pairwise similarity measures are viewed as kernel comparison operations, the record matching problem can be cast as a clustering problem. If two records' similarity exceeds a threshold, they

are considered two variants of the same item. String similarity measures can be classified as either set- or sequence-based, depending on whether ordering information is used or not.

Set-based similarity considers the two strings as independent sets of characters S and T, such as the Jaccard measure – defined as the ratio of the intersection of the sets over the union (i.e., $\frac{|S \cap T|}{|S \cup T|}$). Cosine similarity, borrowed from information retrieval, views both sets as vectors and calculates the angle between the vectors, where a smaller angle indicates higher similarity. Alternatively, asymmetric measures such as *degree of similarity* (i.e., $\frac{|S \cap T|}{|S|}$) may be more appropriate when one string is more important to match than the other.

Sequence-based measures can be generally cast as edit distances. They measure the cost of transforming one ordered string into the other. Typically, the transformation cost is measured by summing the cost of simple incremental operations such as insertion, deletion and substitution.

Hybrids of both set- and sequence-based measures are often used. For example, when the string is a series of words, a sequence-based measure may employed for individual tokens, but the string as a whole may be modeled as a set of tokens [3].

2 Informed Similarity and Record Matching

Library metadata records themselves contain a wide variety of data – personal names, URLs, controlled subject headers, publication names and years. Viewing the list as a database table, each of these columns may have their own notions for what is considered acceptable variation ("Liz" = "Elizabeth"; "Comm. of the ACM" = "CACM"; 1996 $\neq$ 1997). Knowing what type of data exists in a column can inform us of what constitutes similarity and duplication. As such, string similarity measures are usually weighted differently per column.

Certain data types have been studied in much depth. In fact, the need to consolidate records of names and addresses in government and industry pioneered research to find reliable rules and weights for record matching. In set-based similarity, tokens may be weighted with respect to their (log) frequency, as is done in information retrieval models. In sequence-based edit operations, a spectrum of weighting schemes have been used to capture regularities in the data, basically by varying the edit cost based on the position and input. For example, in genomic data, sequences often match even when a whole substring is inserted or deleted; the same is true when matching abbreviations to their full forms. In census data, the initial letters of people's names are rarely incorrect.

Such models need to set parameters such as the cost for each type of edit operation in a principled way. Fortunately, data-driven methods have emerged to learn optimal weights from training data (e.g., [12, 2]).

3 Graphical Formalisms for Record Matching

In recent years, graphical formalisms are becoming popular for record matching. Typically, columns or whole records are viewed as nodes in a graph with edges connecting similar nodes, allowing global information to be incorporated in the disambiguation process. One may assign similarity values to edges and identify cliques of high weights as matching nodes.

A common manifestation of graphical formalisms in disambiguation tasks is in the form of social networks, such as collaboration networks. Social network analysis methods such as centrality and betweeness can be applied. For example in author disambiguation, we may be able to attribute two papers to the same "Wei Wang" when the co-author lists do not have common names, but share names with a third paper – the two nodes are connected by a path through a third node. Yet another work uses network cuts and random walks in the collaboration network of actors to disambiguate names in the Internet Movie Database [7].

Consolidating records using one column of data can sometimes cascade and benefit matching on other columns of the same data. This incremental approach can resolve duplicates when true matching records do not exceed a global similarity threshold before individual fields in the records are merged. Graphical formalisms, such as dependency graphs [4] or conditional random fields [11] nicely model incremental record matching, enabling the propagation of contextual similarity.

Graphical formalisms in the form of generative probabilistic models have also been suggested. In the author disambiguation problem, we can view authors are members of collaborative groups. This model first picks out collaborative groups and then assigns authors within these groups to generate references. We can then run this model in the opposite direction to infer which collaborative group (thus which disambiguated author) is responsible for a particular work [1]. Such graphical models have outperformed methods using pairwise comparisons in accuracy but have yet to demonstrate efficiency on large datasets.

4 Reducing Complexity

Since digital libraries often contain large numbers of records, brute-force pairwise comparison is often infeasible. As of 2005, the estimated number of independent articles and monographs in computer science research alone exceeded 2.2 million [9], an amount unsuited for $O(n^2)$ complexity. (Log) linear time matching algorithms are needed to scale to such large metadata sets.

Observations show that the ratio of true record matches to non-matches is very low – it is very unlikely that two randomly-picked records refer to the same item. Thus, a computationally cheap similarity measure is often used to first separate such implausible matches. These blocking (or canopy) methods map records into a set of blocks in linear time. For example, we can construct a block for all records that have the token "J" and another block for all records that

have the token "Brown". Records that contain both tokens would be members of both blocks. More computationally expensive similarity measures can then be confined to run only within each block, where records have a non-zero probability of matching.

Constructing an optimal block algorithm requires tuning parameters for the proper number of blocks, overlap between blocks and size of the blocks. These parameters can be either rigorously controlled to bound the complexity of the inner comparison [6], or learned from data or sampling [8].

5 Conclusion

Matching problems have matured to become a research issue as early as the 1940s, probably due to the analysis of census data or medical records [6]. Since then, advances have been made both on better theoretical models for weighted matching and proofs for error bounds and optimality.

We believe one promising direction lies with graphical models, which can be tailored to model the underlying structure of the specific record matching scenario. A difficulty in applying these models is in complexity: modeling the structure more accurately requires a more complex graphical model, which in turn creates complexity in the inference procedure. A way of reducing this complexity further would help propel these models for large-scale datasets.

Bringing more knowledge to bear on the problem may also help. Noisy sources such as the web can be seen as a treasure trove of statistical information for matching – if carefully cleaned and utilized. This is especially fruitful for library metadata, as information about authors, titles and publishers are readily available on the web. Motivated by similar approaches in information retrieval research, we have leveraged web search results when disambiguating author names in citation lists [10]. Our study showed that using evidence from such external sources alone can achieve the same disambiguation performance than using the record data itself. We can also ask humans for help directly – by distinguishing which parts of the matching process are easier for humans to solve than machines. The classic Fellegi-Sunter model [12] defines a check zone where uncertain matches are given to human experts to manually check. Similar to approaches used in computer vision, active learning based on manual disambiguation can help create more accurate matching systems. Elusive, domain-specific matching knowledge may be easier to capture by having human experts solve example problems rather than asking them to code explicit rules.

In conclusion, it is unclear whether advances in record matching have kept up with the pace at which information is becoming widely available today. In the world of digital libraries, metadata inconsistencies are still a constant barrier to locating and collating knowledge, which end users and reference librarians have had to adapt to. In some cases, humans resort to using external sources of information to (in)validate a possible match. As more information becomes web-accessible, we expect mining such external sources for knowledge will play an increasingly useful role in matching problems. We believe that incorporating

such external yet accessible knowledge gathering as an active component of matching algorithms will be a valuable research direction.

References

[1] Indrajit Bhattacharya and Lise Getoor. A latent dirichlet model for unsupervised entity resolution. In *SIAM International Conference on Data Mining*, pages 47–58, April 2006.

[2] Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39–48, August 2003.

[3] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Information Integration on the Web (IIWeb)*, pages 73–78, August 2003.

[4] Xin Dong, Alon Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *ACM SIGMOD International Conference on Management of Data*, pages 85–96, June 2005.

[5] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(1):1–16, January 2007.

[6] Mauricio A. Hernández. *A Generalization of Band Joins and The Merge/Purge Problem*. PhD thesis, Columbia University, March 1996.

[7] Bradley Malin, Edoardo Airoldi, and Kathleen M. Carley. A network analysis model for disambiguation of names in lists. *Computational and Mathematical Organization Theory*, 11(2):119–139, July 2005.

[8] Matthew Michelson and Craig A. Knoblock. Learning blocking schemes for record linkage. In *National Conference on Artificial Intelligence (AAAI)*, July 2006.

[9] Vaclav Petricek, Ingemar J. Cox, Hui Han, Isaac Councill, and C. Lee Giles. A comparison of on-line computer science citation databases. In *European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 438–449, September 2005.

[10] Yee Fan Tan, Min-Yen Kan, and Dongwon Lee. Search engine driven author disambiguation. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 314–315, June 2006.

[11] Ben Wellner, Andrew McCallum, Fuchun Peng, and Michael Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 593–601, July 2004.

[12] William E. Winkler and Yves Thibaudeau. An application of the Fellegi-Sunter Model of record linkage to the 1990 U.S. Decennial Census. Technical Report RR91/09, U.S. Bureau of the Census, 1991.