CoBib: An Architecture for a Collaborative Database

by

Beth Trushkowsky

Department of Computer Science
Duke University

April 25, 2007

Abstract

The goal of CoBib is to allow affinity groups to effectively collaborate to maximize the searching and browsing utility of an academic paper database. The CoBib system will facilitate the process of surveying literature in a specific field by using the community's annotations and referrals. I am developing a database architecture for CoBib that provides users within research communities the means to collaboratively index and annotate citations. This extensible architecture is a novel solution that is interoperable with existing data formats and systems and incorporates recommendations gathered from the community for the discovery of new citations. The construction of CoBib raises several important questions regarding data integrity and utility. One such question is how to determine whether two citations refer to the same paper. This problem is generally referred to as *object coreference*. I discuss several current approaches to the object coreference problem. I additionally describe issues related to the back-end and front-end design: how to choose an appropriate database for a large set of records as well as the best means for users to query that database.

1. Introduction

Online digital libraries facilitate researchers' pursuit of academic papers because they provide users with an interface into potentially large databases using familiar Boolean search techniques and/or restrictive queries. There are several prominent existing systems, namely CiteSeer and Google Scholar, that crawl the web for academic papers and allow users to search their respective databases. What's missing from these current systems is user collaboration and interaction with the site. These features are important in academic research because people often have substantial knowledge about relationships between papers and can offer meaningful opinions about such resources. Accessing this alternate semantic information would increase the users' productivity. For example, a user new to a particular field should be able to quickly find the prominent papers and people in that field. With user-provided annotations and paper relationships, it will be easier to obtain a literature review of a particular subject.

My goal was to create an effective way for affinity groups in a small academic community to effectively collaborate to maximize the searching and browsing utility of an academic paper database. CoBib is a system that enhances current implementations of citation databases by incorporating user interaction via user-provided data, user tagging, and recommending. Recommending is accomplished explicitly through direct suggestions to particular people or interest groups, and implicitly through collaborative filtering algorithms.

CoBib's construction several important design questions concerning data storage and manipulation as well as the user interface. One of the primary obstacles facing developers of citation databases is maintaining the *integrity* of the system. In this context, integrity means that each citation in the database refers to a unique academic paper. Users will undoubtedly employ varying citation styles, and thus different citations that refer to the same paper must be reconciled. This problem is more broadly

defined as *object coreference*. Another essential component of a system's efficacy involves the users' interaction with that system. As a modern online application, CoBib is expected to have an intuitive interface and allow users to retrieve results accurately and quickly.

I have found that string matching algorithms are an effective starting approach to the citation matching problem. My implementation detects coreferenced citations that contain misspellings or differing completeness. I use a Zebra server to house the citations and a relational database to maintain information about the relationships between users and citations. The Zebra server scales well to a large dataset, and offers advanced querying options backed by Library of Congress guidelines. The current user interface uses word-completion suggestions to expedite effective searching, and attempts to hide less frequently used elements, like citation correction. Users have appreciated these features, but would benefit from more intelligent suggestion methods than simple prefix-matching.

This paper describes the characteristics of exiting academic paper databases, highlighting their strengths and weaknesses. I follow that discussion with an outline of the design principles that guided CoBib's development. Those principles are utilized in the discussion of the database and user interface designs. I explore citation matching by summarizing three different approaches to the object coreference problem before describing my implementation. The following section describes the recommending process via collaborative filtering and tagging. I conclude with examples using several of CoBib's functionality and discuss future work.

2. Related Work

2.1 Academic Paper Databases

Existing online repositories of research papers have varying strengths and weaknesses. Here I discuss several of the prominent systems and their respective attributes.

CiteSeer [9] crawls the web for academic documents specifically in the fields of computer science and engineering. This focus is advantageous because it emphasizes depth rather than breadth, increasing a user's chances of finding the resources he is looking for. CiteSeer also uses autonomous citation matching, which involves parsing documents for works cited and indexing these results along with the original paper. Cross-referencing papers in this manner allows the user to easily discover related resources. While CiteSeer allows users to make comments on particular papers, these comments are generally directed towards the website's administrator rather than other users. Thus users don't necessarily gain an understanding of their peers' opinions about the papers. Another possible weakness is CiteSeer's corrections policy. Any modification made by a user has to be reviewed and thus may limit the user's inclination to make corrections. Additionally, CiteSeer lacks an intuitive user interface: the user is bombarded with information and may have trouble discerning the pertinent material.

Google Scholar [12] is similar to CiteSeer in that it crawls the web for academic papers. It does not, however, limit its scope to any particular discipline. The system does have the advantage of the simple and familiar Google interface, but that simplicity may have its costs. Google Scholar offers no user interaction with its data, either through modifications or annotation. There is no concept of relevant keywords associated with the papers, nor does it specify how it defines "related articles."

However, Google Scholar does have the advantage of significant processing power and incomparable speed.

Rexa [31] is a project from the University of Massachusetts at Amherst that began its development around the same time as CoBib. This site explores object coreference in citations and authors, and their future work will extend such reconciliation to broader fields like grants and conferences. Rexa's automated indexing and reconciliation demonstrates a significant endeavor in the field, but their method is prone to errors. There are duplicate citations in their database, papers are attributed to the wrong authors, or biographical information is incorrect. Rexa appears to allow users to correct BibTeX data, but it is unclear whether these changes are eventually reflected in their main database. One of Rexa's strengths is providing users with the capability to tag papers as a bookmarking tool. This feature assists users in maintaining their own groupings and paper associations for future work. Rexa does not, however, show tags that other users have placed on a particular citation. Thus the system lacks any community interaction.

Karger's Haystack project [15] concerns the broader field of personal information management (PIM). Like academic paper databases, PIM involves the conglomeration of information from disparate sources in order to maximize the utility of that information. This project sought to break the boundaries of software-dependent information, such as email addresses to an email client, and merge that information into customized workspaces. Haystack borrows ideas from the Semantic Web [1], namely that the distribution of "information" should not be limited by the medium in which it is expressed. CoBib realizes this concept through my understanding of my users' intellectual contributions to the community. Thus I incorporate textual data from many sources via user-submitted citations, as well as another layer of semantics via users' comments and annotations.

2.2 Collaborative Work

Other systems have been developed that take advantage of a community's ability to improve the relevancy and value of the data in question. Freyne produced a Web search system that will modify the ranking of search results to adhere to the community's interests [5]. Similarly, Kautz developed a system that will search the Web utilizing a user's social network [18]. The objective in this system was to garner trusted information via trusted experts. Both Web search schemes illustrate the usefulness of research in a collaborative environment: a user's community can facilitate the acquisition of relevant information.

3. Principles

3.1 Goals

CoBib is designed to fulfill particular academic needs and desires in a novel way. Primarily, the system will allow users to obtain a literature review of a particular subject. A user new to a certain area can quickly discover the prominent papers and people in that area. The community can also assist its members through tagging, annotating, and recommending; this promotes the easy exchange of ideas. The recommendation process also provides a view into papers that are currently important or "hot." Papers that are often explicitly recommended demonstrate a topic that is frequently discussed in current literature. Implicit recommendations highlight papers dealing with hot topics because they represent ideas that a lot of users are currently interested in. By keeping the CoBib community restricted to a small academic neighborhood, I can be more comfortable trusting users' recommendations, as well as other user-contributed data. CoBib will also allow users to maintain a bookshelf of

citations with customized groupings. These groups can be used to organize citations for a particular paper in progress, and can be shared with other users for possible collaborations.

CoBib will focus on the two key methods of data querying, searching and browsing, as well as incorporating ideas from Web 2.0 [29]. Searching is a direct query into the database of citations, meaning that the user is looking for a particular paper or author. Browsing is more indirect and involves the user discovering new citations from known references or keywords. Browsing becomes more effective when we allow users to make meaningful contributions through tagging and annotating. User collaboration and interaction with a site is a major component of Web 2.0. Not only is website content more fluid, it is expected to be quickly delivered. Thus I propose using asynchronous processing to expedite processing of users' requests.

3.2 Design Principles

The data stored and managed by CoBib will adhere to a number of information standards that facilitate its use in the academic community. These standards are all openly published, freely available specifications that can be used with few (if any) restrictions. The use of established methods and protocols ensures the reliability, portability, and interoperability of CoBib's features.

A Zebra server [13] that implements the z39.50 protocol maintained by the Library of Congress [23] indexes the bibliographic records in CoBib's database. This protocol defines communication between a client's request for information from a remote computer database. CoBib also uses z39.50 to seamlessly query the myriad library and university databases that use the protocol as well [7]. By keeping the syntax of the client's search query independent of the server's database structure, z39.50 allows the client to remain blind to the server's inner workings while still being able to make reliable, intuitive queries. The semantics for anticipated common searches are provided by the

ISO registered Bath Profile [30], a commonly used profile developed for use in library applications. Thus the implementation of a search for author keyword can be formally defined by the profile to garner the most appropriate results from the database.

The structure of the bibliographic records is defined by the Metadata Object Description Schema (MODS), from the Library of Congress [21]. This XML schema was developed to adhere to the MARC 21 standard [20] for representing bibliographic records in a machine-readable form, as well as to provide the readability and portability of using XML to store textual information. CoBib maintains its database using the MODS schema to ensure record uniformity and integrity. In addition to receiving MODS XML citations, CoBib accepts bibliographic records from its users in the BibTeX, and EndNote formats [25,4]. Members of the academic community commonly maintain these bibliographic files for accessible use in academic papers; uploading to CoBib will require no extra interfacing on the part of the user. Additionally, CoBib allows its users to download citations in a familiar format to be used in compiling future work.

The citations that CoBib possesses a document for will be available to users in the Portable Document Format for Archives (PDF/A) [16]. This file format, used for long-term digital archiving, is platform independent and widely used to view academic papers online. Using PDF/A ensures that the documents stored by CoBib will be self-contained: they do not need to reference any external sources in order to be rendered. This self-containment secures the documents' usability over time.

4. Design

4.1 Databases

There are two fundamental types of data being stored in CoBib: (1) the bibliographic record and (2) the user and citation profiles and relationships. Differentiating between these two types of data assists in choosing the most

appropriate data storage method for each. Both databases will be used in the user's everyday interactions with the CoBib system, the most common of those actions will be uploading and searching for citations.

One of the most important features of a database that will experience a great amount of queries is the processing time. I implemented a SAX parser that could search for a particular string by iterating through the elements of an XML record. As the number of records increased, I found that the processing time increased dramatically, and was thus not practical for a large database of citations. The SAX parser implementation also did not support queries more complex than substring matching.

I chose to use a Zebra server to interact with the XML in order to index the citations for retrieval upon the user's request. This Zebra database server is efficient and reliable because it automatically indexes its records and scales well to large datasets. It is a more appropriate means to interface the number of records that CoBib will hold, and can be used to provide tailored searching capabilities.

The CoBib system also utilizes a MySQL server to house information connecting citations to users. This user and citation relationship information refers to the data accumulated by the users' interaction with the citations, may it be through uploading, viewing, tagging, annotating, editing, or recommending citations. This data is stored in a relational database, with most tables describing what user x did with citation y. The purpose of this database is to provide a means for CoBib to maintain the relationship amongst these entities in a way that facilitates retrieving and cross-referencing information that is inherently nonhierarchical.

The process of searching and browsing citations is provided by the Zebra server and the relational database, respectively. To search directly for a citation, users can perform complicated queries on specific bibliographic fields using Boolean operators and/or regular expressions. This query is sent to the Zebra server, which will return a ranked result list. Alternatively, users may choose to browse for citations by viewing

tags or recommendations. The relational database allows the user to discover new citations by providing the relationship between records through which the user will navigate.

When a user uploads a file of citations, may it be from a bibliographic file, form fields, or another Web database, the following process occurs (See Figure 1): (1) web form data is converted to a BibTeX entry to facilitate transformation into XML, (2) the BibTeX/EndNote data is converted [28] into MODS XML, (3) citation matching algorithms are used to determine if the new citation refers to an existing record in the database, (4) if there is an existing record, the content of that citation is updated with any new information, otherwise new entries are created in the relational database and Zebra server. I determine whether a record exists by using string-matching algorithms on the primary bibliographic fields.

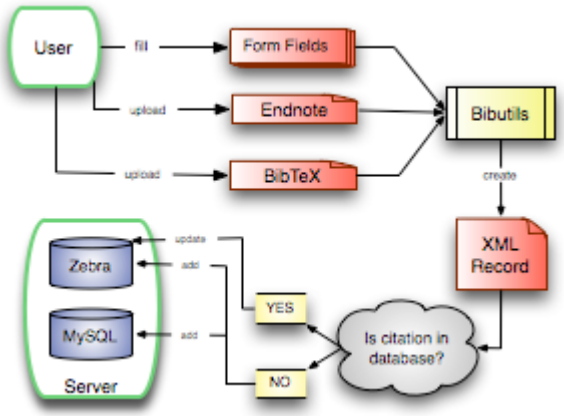

Figure 1: Procedure for uploading a citations file. The bibliographic data is converted to an XML file and run through citation matching algorithms to determine if that record already exists. The server is then updated to reflect either a new citation or new information for an existing record.

4.2 Citation Matching

The utility of CoBib's citation database may be determined by the quality of the citations—citation matching is imperative in maintaining the accuracy of the database. By finding an effective automated means to decide whether two citations refer to the same paper, CoBib can be able to remove duplicate entries and retain the maximum

amount of data available for a particular record. Citation matching refers to the object coreference problem, also known as *identity uncertainty* and *reference reconciliation*, amongst other terms. The problem entails determining if two different object descriptions refer to the same object. In this context, the objects are citations. The disparity can be as simple as a typographical error, or may involve incorrect or incomplete information. This problem is an active area of research, and I describe several approaches. I conclude with our current implementation.

The approach described in Pasula [24] uses relational probability models [26] (RPM) to reason about object identity. In their RPM, the authors specify a set of classes representing a citation and its components: *Citation, Author, Paper*, *AuthorAsCited*, etc. Complex attributes describe functional relations between classes, and simple attributes denote probabilistic dependencies. Once an RPM has been created, it can be converted into an equivalent Bayesian network. The simple attributes will become nodes in the graph, and their parents and grandparents will be instance dependencies and citation objects, respectively. Given two citations, C_1 and C_2, in a Bayesian network, the probability that they refer to the same paper can be determined by the number of shared attributes. An approximation of probability is computed with Markov chain Monte Carlo [10]. The authors found this technique to be effective in a small data set, but it did not scale up well. As the number of papers increases, the chance that any two particular clusters have many common attributes will be low. They propose running a cheap distance metric over a large dataset to partition the data into subsets that are more likely to have matching attributes.

Dong's technique [3] also takes advantage of the relationships between instances of objects. They additionally address the potential issue that a particular citation may have very little information. Their model defines a set of classes, each with a set of attributes. Class instances represent particular citations, and their goal is for each instance to correspond to a single real-world entity and vice versa. Attributes come in

two varieties: *atomic* attributes describe simple values, while *association* attributes describe links to other instances. The reconciliation algorithm has three main objectives: (1) use associations between references to assist reconciliation decisions, (2) propagate reconciliations to neighboring references, and (3) enrich references after reconciliation. Thus each reconciliation will update information in the entire domain. The algorithm uses a dependency graph, whose nodes denote similarities between pairs of references, and whose edges denote dependencies between reconciliation decisions. The similarity score of a pair of elements is the sum of the scores of its real-valued incoming neighbors and the scores of its strong and weak Boolean-valued neighbors.

A different approach involves string matching algorithms. Cohen [2] describes three types of algorithms: edit-distance, token, and hybrid. In an edit-distance algorithm, the number of edit operations needed to transform one string into the other determines the similarity between them. Token-based algorithms use matching word frequencies within strings to assign scores.

CoBib will employ the hybrid approach, in which a combination of edit-distance and token-based methods results in an algorithm that computes the frequency of similar tokens, with the similarity of tokens computed by the edit-distance algorithm. The advantage of CoBib's bibliographic record storage is that specific fields in each record have been delimited in the MODS XML. This delimitation facilitates tailoring the citation matching by applying appropriate string matching algorithms to individual fields. I have chosen to use the hybrid Jaro-Winkler [17] and hybrid Monge-Elkan [22] algorithms for author and title fields, respectively. Jaro-Winkler is an edit-distance algorithm that adjusts its score to favor matching prefixes, which is applicable to matching misspelled or incomplete names of authors. Monge-Elkan favors matching substrings, making it an appropriate choice for title fields. The figure below illustrates the current implementation of citation matching using the string algorithms. The

performance of the algorithms used will improve over time as users provide feedback on the correctness of matches that the system has made.

Figure 2: Citation matching task after an upload. The user has uploaded a duplicate citation with a misspelling in the author's name.

4.3 User Interface

CoBib's utility is largely influenced by users' ease in interacting with it. An intuitive user interface allows the user to quickly find the information they seek, and is thus part of the design process. *User-centered design* is a design philosophy that focuses on the needs and wants of the end-user of a resource throughout the design process of that resource. In designing CoBib, my goal was to decrease processing time and increase productivity. I used asynchronous processing via AJAX [8] with word suggestions to facilitate users' searching and tagging. The figure below illustrates a user successfully finding an author's name after beginning a search.

Another user design issue involves active user participation. Users play a crucial role in maintaining the integrity of the database through their citation reconciliations and corrections. To increase the likelihood that users will perform these tasks, I propose that all actions preceding that task will be automated. By making the overall procedure easier, users will be more inclined to put forth minimal effort. This "give a lot, take a little" mentality can be further achieved by allowing users to upload documents by

dragging them onto the CoBib site, rather than having to browse through directories. Users are then asked to reconcile conflicts the next time they log in.

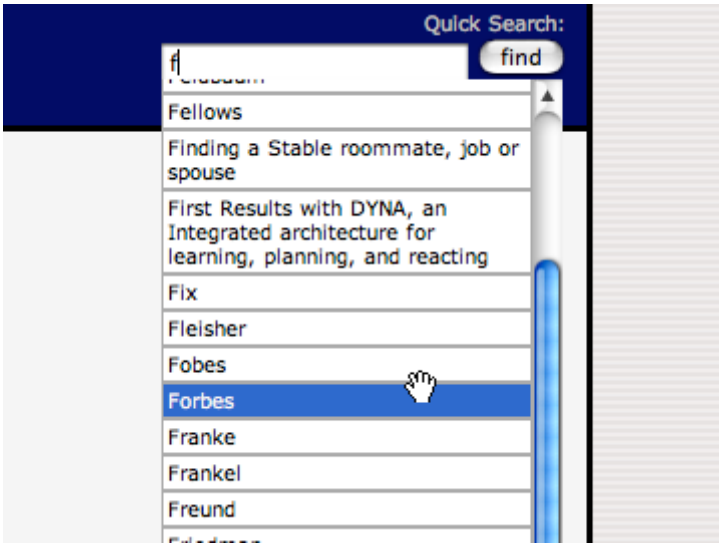

Figure 3: Immediate search suggestion using AJAX

5. Recommending

5.1 Collaborative Filtering

CoBib's recommendation process consists of both explicit and implicit techniques. For a direct recommendation, users are able to suggest a citation to other specific users, or to users who are part of a given group or have certain interests. Additionally, CoBib will use collaborative filtering [14] to implicitly perform recommendations: the actions of all the users in the community will influence the citations suggested for a particular user.

Both recommendation methods rely on determining what a user's interests entail. This data will be garnered both explicitly and implicitly as well. The former involves direct input from the user, either through a supplied list of interest keywords or implied keywords derived from the user's research area. Tracking user actions will allow for implicit gathering of user actions. Here an action is defined as viewing, uploading,

tagging, and annotating citations. By logging which topics a particular user seems to prefer, CoBib will provide meaningful recommendations.

5.2 Tagging

Tagging is officially referred to as the construction of folksonomies [32], and is used to describe user-generated keywords applied to web content. The major benefits of folksonomies are twofold. As implemented by Rexa, tagging can be used to bookmark and group citations for personal organization. CoBib provides the additional utility of social tagging: by connecting users' tagging activities with the rest of the community, everyone can benefit from their peers' annotations. Using social tagging increases success in citation browsing because the community provides a means for grouping relevant information.

Social tagging can lead to what has been called the *vocabulary problem* [6]. In this context, the problem arises when users write tags using different words that refer to the same idea. A simple example would be the tags "databases" and "database systems." While they likely describe related ideas, the citations would not be grouped together because of the differing vocabulary. Consistency in vocabulary is crucial to the effectiveness of social tagging. To combat this problem, CoBib uses asynchronous processing to offer tag suggestions to users as they are typing. Thus if a similar tag exists in the database, users will choose the one that fits their ideas, rather than creating a new tag that may essentially duplicate existing tags.

6. Use Cases

6.1 Searching by Tag

Tagging is one of the most recent trends in collaborative research and classification. One of CoBib's most prominent features is its collaborative tagging

mechanism that allows users to recommend papers to one another by marking them through a shared language. As a research tool, collaborative tagging systems enhance research by giving broader treatment to papers that might only appear in narrowly defined searches [32]. Tagging lends intuition to the otherwise automated process of the generic search.

For example, a CoBib user looking for a paper on education has the option of searching through papers tagged *undergraduate* on the CoBib site. The user can further refine this search by searching for papers tagged *undergraduate* by users in a specific interest group, such as *education* (See Figures 4 and 5). This ability to search for a specific topic as it relates to a broader group improves the quality of the user's research. Instead of seeing all papers with some reference to undergraduates, a search of tags allows the user to narrow his results to papers with reference to undergraduates as it relates to those interested in education.

Figure 4: A user search for the tag *undergraduate* from the interest group *education*

Figure 5: The results of the search by tag, showing matching citations and the users who tagged them

6.2 Uploading a Paper

Each time a user uploads a paper, he will have the option of correcting the citation information being uploaded to the site. For example, if a user wants to upload a paper to the site, he can simply drag and drop the paper onto the citation extractor.

Once CoBib has generated the citation information, the user will be asked to confirm and/or correct this information. The substantial amount of automation performed by the system will increase the likelihood of the user contributing in the final steps in the process of generating a citation record.

6.3 The Profile Pane

The profile pane is on every page of the CoBib website, allowing the user to easily access commonly used functions. Figures 4 and 5 illustrate these features: (1) recommendations for the user, with a count of unviewed entries, (2) the bookshelf, a site for the user to organize his citations by subject or current project, (3) the current list of reconciliation tasks, i.e. the list of citations that the user attempted to upload that have potential matches for existing records, (4) the upload file function to contribute more citation records to the database, and (5) the area to edit the user's profile, which includes personal information as well as the list of interests to be used in the recommendation process.

7. Conclusions and Further Work

The most prominently occurring obstacle in developing a system like CoBib involves maintaining fast processing time while the amount of data increases dramatically. My initial implementation to query the citations database using a SAX parser demonstrated the difficulty in accessing data stored in a hierarchical format like XML. The decision to switch to a Zebra server was well-founded not only because of its scalability, but also because Zebra understands more complex queries than matching substrings. However, I have not taken advantage of citation matching when querying the Zebra server. If this addition is possible, it may increase the users' contentment with the results they obtain when running searches. Object coreference techniques may also

benefit search success if used with the AJAX word-suggestion, rather than a simple prefix matching.

Another slow functionality is the cascade of processes that occur when a user uploads a new citation. CoBib currently extracts keywords for word-suggestions and runs the citation matching algorithms for each citation while the user is waiting for the system to finish. This waiting time is a hindrance and may discourage users from uploading citations. I propose running alternate threads that process the uploaded citations without necessitating a response from the user.

Other issues involve the reconciliation and retention of data in the database. CoBib currently assumes that a newly uploaded citation that refers to the same paper as an existing citation should be removed. However, the new version may be more accurate or complete. I will alter the interface to allow users to pick and choose fields from both citations until the most complete version is stored in the database. There may be a time, though, when citations should not be modified further by users. For example, a particular citation may have been viewed frequently over a long period of time without any modifications, and then a user changes its values. The system should be more inclined to trust the number of satisfied viewings than the updated values. Thus a more complex model of user interaction should be implemented that takes such trends into consideration.

As user studies are performed in the computer science department, I will gain more insight into CoBib's efficacy as a searching and browsing tool.

References

[1] Berners-Lee, Tim, Hendler, James, Lassila, Ora. *A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*. ScientificAmerican.com, May 2001. <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>.

[2] Cohen, William, Ravikumar, Pradeep and Fienberg, Stephen (2003): *A Comparison of String Metrics for Matching Names and Records* in KDD Workshop on Data Cleaning and Object Consolidation.

[3] Dong, X., Halevy, A., and Madhavan, J. *Reference reconciliation in complex information spaces*. In Proceedings of the 2005 ACM SIGMOD international Conference on Management of Data (Baltimore, Maryland, June 14 - 16, 2005). SIGMOD '05. ACM Press, New York, NY, 85-96.

[4] *Endnote: Bibliographies made easy*. The Thomson Corporation, 2006. <http://www.endnote.com>.

[5] Freyne, J., Smyth, B., Coyle, M., Balfe, E., and Briggs, P. 2004. *Further Experiments on Collaborative Ranking in Community-Based Web Search*. Artif. Intell. Rev. 21, 3-4 (Jun. 2004), 229-252.

[6] Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. *The vocabulary problem in human-system communication*. Commun. ACM 30, 11 (Nov. 1987), 964-971.

[7] *Gateway to Library Catalogs*. 2006. <http://www.loc.gov/z3950/gateway.html>.

[8] Garrett, J. *Ajax: A New Approach to Web Applications*. <www.adaptivepath.com/publications/essays/archives/000385.php>.

[9] Giles, C.L., Bollacker, K., and Lawrence, S., *CiteSeer: An Automatic Citation Indexing System*, Digital Libraries 98: Third ACM Conf. Digital Libraries, ACM Press, New York, 1998, pp. 89-98.

[10] Gilks, W.R., Richardson, S., Spiegelhalter, D.J. *Markov chain Monte Carlo in practice*. Chapman and Hall, London, 1996.

[11] Glover, E. J., Lawrence, S., Gordon, M. D., Birmingham, W. P., and Giles, C. L. 2001. *Web Search---Your Way*. Commun. ACM 44, 12 (Dec. 2001), 97-102.

[12] *Google Scholar*. <http://scholar.google.com>.

[13] Hammer, Sebastian, Dickmeiss, Adam, et. al. *Zebra – User's Guide and Reference.* August 2006. <http://www.indexdata.dk/zebra/doc>.

[14] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. 2004. *Evaluating collaborative filtering recommender systems*. ACM Trans. Inf. Syst. 22, 1 (Jan. 2004), 5-53.

[15] Huynh, David, Karger, David R., Quan, Dennis, Sinha, Vineet. *Haystack: A Platform for Creating, Organizing and Visualizing Semistructured Information* (2002).

[16] *ISO 19005-1:2005: Document management—Electronic document file format for long-term preservation—Part 1: Use of PDF 1.4 (PDF/A-1)*, International Organization for Standardization, Geneva, Switzerland.

[17] Jaro, M. A. *Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida*. Journal of the American Statistical Association 84:414–420, 1989.

[18] Kautz, H., Selman, B., and Shah, M. 1997. *Referral Web: combining social networks and collaborative filtering*. Commun. ACM 40, 3 (Mar. 1997), 63-65.

[19] Lawrence, Steve, Giles, C.L., Bollacker, Kurt. *Autonomous Citation Matching*, Proceedings of the Third International Conference on Autonomous Agents, Seattle, Washington, May 1–5, ACM Press, New York, NY, 1999.

[20] *MARC 21 concise format for authority data*. Library of Congress: Network Development and MARC Standards Office, 2000. <http://purl.access.gpo.gov/GPO/LPS42660>.

[21] *Metadata Object Description Schema (MODS).* <http://www.loc.gov/standards/mods/>.

[22] Monge, A., and Elkan, C. *The field-matching problem: algorithm and applications.* In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996.

[23] NISO, ANSI. *Information retrieval (Z39.50): application service definition and protocol specification: an American national standard.* Bethesda, MD: NISO Press, 2003.

[24] Pasula, Hanna M., Marthi, Bhaskara, Milch, Brian, Russell, Stuart, and Shpitser, Ilya, *Identity Uncertainty and Citation Matching*. Advances in Neural Information Processing Systems 15 (NIPS 2003). Cambridge, MA: MIT Press.

[25] Patashnik, Oren. *BibTEXing*. 8 February 1988.

[26] Pfeffer, A. *Probabilistic Reasoning for Complex Systems*. PhD thesis, Stanford, 2000.

[27] Polack-Wahl, J. A. and Anewalt, K. 2006. Learning strategies and undergraduate research. *In Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06. ACM Press, New York, NY, 209-213.

[28] Putnam, Chris. *Bibutils: Bibliographic conversion utilities.* October 2005. <http://www.scripps.edu/~cdputnam/software/bibutils/>.

[29] O'Reilly, T. *What is Web 2.0? Design patterns and business models for the next generation of software*. 2005. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.

[30] *The Bath Profile: An International Z39.50 Specification 1.0, April for Library Applications and Resource Discovery.* 1999. Ed. Lunau, Carroll, Miller, Paul, and Moen, William. <http://www.ukoln.ac.uk/interop-focus/activities/z3950/int_profile/bath/draft/stable1.html>.

[31] Wellner, Ben, Callum, Andrew, Peng, Fuchun, Hay, Michael. *An Integrated, Conditional Model of Information Extraction and Coreference with Application to Citation Matching*. Conference on Uncertainty in Artificial Intelligence (UAI), 2004.

[32] Wu, H., Zubair, M., and Maly, K. *Harvesting social knowledge from folksonomies*. In Proceedings of the Seventeenth Conference on Hypertext and Hypermedia (Odense, Denmark, August 22 - 25, 2006). HYPERTEXT '06. ACM Press, New York, NY, 111-114.