CoBib: Collaborative indexing and annotation of bibliographic citations

Beth Trushkowsky
Computer Science Division
University of California
Berkeley, CA 94720
trush@eecs.berkeley.edu

Dave Stecher
Department of Computer Science
Duke University
Durham, NC 27708
dms27@duke.edu

Jeffrey Forbes
Department of Computer Science
Duke University
Durham, NC 27708
forbes@cs.duke.edu

ABSTRACT

CoBib facilitates the process of surveying literature by using a community's actions, annotations, and referrals. The database architecture for CoBib provides users within research communities the means to collaboratively index and annotate citations by supporting both searching and browsing behavior. The system enables users to learn about research through explicit and implicit recommendations. This poster describes the principles and architecture of CoBib and our work towards effectively sharing references among research communities.

Categories and Subject Descriptors

H.3.7 [**Digital Libraries**]: User issues

General Terms

Management, Human Factors, Design

The goal of the CoBib project is to design, implement, and test a web database that enables collaborative indexing and annotation of citations within a research community. This system facilitates the structural and social elements of research: a survey of literature in a specified field, as well as discussion of the prevalent researchers or themes in that field. The database stores citations that refer to research papers. Researchers locate citations by searching, a direct query for a particular citation, or by browsing, an indirect survey of existing records.

CoBib adapts some of the functionality of other systems designed for searching for technical papers: a web interfaee to papers (e.g. GoogleScholar); and keyword tags for citations (Rexa[7]). CoBib houses citations, rather than referring to papers on the web, while allowing for a variety of annotation for each citation and using item-based collaborative filtering[5] to generate recommendations. Other systems have been developed that take advantage of a community's ability to improve the relevancy and value of the data in question[6]. These Web search schemes illustrate the usefulness of research in a collaborative environment: a user's community can facilitate the acquisition of relevant information. Like other systems, CoBib adheres to a number of information standards that facilitate its use in the academic community. This architecture is a novel solution for effectively sharing references among and within research communities.

CSCW'08, November 8–12, 2008, San Diego, California, USA.

DESIGN

Databases

There are two fundamental types of data being stored in CoBib: (1) the bibliographic record and (2) the user and citation profiles. A Zebra server will interact with the XML bibliographic records in order to index the citations for retrieval upon the user's request[4]. The user and citation profile information refers to the data accumulated by the users' interaction with the citations through uploading, viewing, tagging, annotating, editing, or recommending citations. This information is stored in a relational database. The purpose of this database is to provide a means to maintain the relationship amongst these entities in a way that facilitates retrieving and cross-referencing information that is inherently nonhierarchical.

Citation Matching

The utility of CoBib's citation database may be determined by the quality of the citations. Different citations that refer to the same paper must be reconciled. This problem is more broadly defined as object coreference[9]. CoBib uses a hybrid edit-distance and token-based approach for matching citations[2]. In an edit-distance algorithm, such as Levenstein and Monge-Elkan, the number of edit operations needed to transform one string into the other determines the similarity between them. Token-based algorithms use word frequencies within the strings. CoBib uses an algorithm that computes the frequency of similar tokens, with the similarity of tokens computed by the edit-distance algorithm. The performance of the algorithms used will improve over time as users provide feedback on the correctness of matches that the system has made. By using an effective automated means to decide whether two citations refer to the same paper, CoBib will be able to remove duplicate entries and retain the maximum amount of data available for a particular record.

Generating Recommendations

CoBib's recommendation process consists of both explicit and implicit techniques. For an explicit recommendation, users will be able to suggest a citation to other specific users, or to users who are part of a specified community. Similarly, users can tag citations to recommend them to groups. CoBib uses item-based collaborative filtering[5] to implicitly perform recommendations: the actions of all the

users in the community will influence the citations suggested for a particular user. We use Taste [8], an open source Java recommendation engine, as our recommender server. The strength of a user's rating for citation is based on the following actions: (1) confirmed authorship, (2) uploading, (3) viewing, and (4) community actions.

The recommendation methods rely on determining what a user's interests entail. This data will be garnered both explicitly and implicitly. 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 community. Tracking user actions will allow for implicit gathering of user actions: viewing, uploading, tagging, and annotating citations.

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. Social tagging can lead to the vocabulary problem[3], when users write tags using different words that refer to the same idea. To combat this problem, CoBib offers tag suggestions. Thus if a similar tag exists, users will choose the one that fits their ideas, rather than creating a new tag that may duplicate existing tags.

A novel use of CoBib is to discover potential research communities. Duke maintains an online Faculty Database System with lists of faculty publications in a somewhat standardized form. The web scraper downloads these citations for faculty in the physical sciences and engineering into BibTeX files that are then uploaded into CoBib. CoBib generates a table mapping each author to their coauthors. Using this data, we can generate a coauthorship graph where faculty authors are vertices, and authors are connected by an edge if and only if they coauthored a paper or share a coauthor. The generated graph has 319 vertices and 5,519 edges. Using community structure discovery algorithms[1], we can find communities within and across disciplines. We hypothesize that utilizing information about a user's research community, whether explicitly stated or inferred by their actions and the structure of the coauthorship network, will help provide more meaningful recommendations.

User Interface

CoBib's utility is largely determined by users' contributions to the database in terms of citations, annotations, and reconciliation activity. An intuitive user interface allows the user to quickly find the information they seek, and is thus part of the design process. One of CoBib's design goals was to decrease processing time and increase productivity. We used asynchronous processing via AJAX and background processing for uploading citations, generating recommendations, and other computationally expensive tasks.

CURRENT STATUS

The core features of CoBib described above have been implemented. Scraping citations from webpages is currently done with a separate program, but the process can be integrated with CoBib. The next stage in CoBib development will be to improve the performance of the citation matching and collaborative filtering algorithms to generate more accurate matches for each of their respective tasks, as well as to refine the user interface such that the user experience is straightforward and intuitive. An upcoming pilot test of CoBib within Duke University's Computer Science department will facilitate the execution of this stage. User studies will evaluate the effectiveness of the system across and between research communities. These user studies will help assess our progress to project goals of broad departmental usage, ease of use, mixed user interaction, and portability.

ACKNOWLEDGMENTS

This work was supported in part by NSF Award CNS-0722288 and a CRA CREU award.

REFERENCES

1. Clauset, A, Newman, M.E.J., and Moore, C, Finding community structure in very large networks. *Phys. Rev. E* 70:6, 2004.
2. Cohen, William, Ravikumar, Pradeep and Fienberg, Stephen. A Comparison of String Metrics for Matching Names and Records, In *KDD Workshop on Data Cleaning and Object Consolidation*., 2003.
3. Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. The vocabulary problem in human-system communication. *CACM* 30, 11 (Nov. 1987), 964-971.
4. Hammer, Sebastian, Dickmeiss, Adam, et. al. *Zebra – User's Guide and Reference.* August 2006.
5. 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.
6. Kautz, H., Selman, B., and Shah, M. 1997. *Referral Web: combining social networks and collaborative filtering*. Commun. ACM 40, 3 (Mar. 1997), 63-65.
7. Mann, Gideon, Mimno, David, McCallum, Andrew. Bibliometric Impact Measures Leveraging Topic Analysis *Joint Conference on Digital Libraries* (JCDL) 2006.
8. Owen, Sean R. *Taste: Collaborative Filtering for Java.* <http://taste.sourceforge.net>
9. 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.