Learning Metadata from the Evidence in an On-Line Citation Matching Scheme

Isaac G. Councill[1] Huajing Li[2] Ziming Zhuang[1] Sandip Debnath[2]
Levent Bolelli[2] Wang-Chien Lee[2] Anand Sivasubramaniam[2] C. Lee Giles[1]

[1]The School of Information Sciences and Technology
[2]Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
igc2@psu.edu, {huali,debnath,wlee,anand}@cse.psu.edu,
{zzhuang,giles}@ist.psu.edu, bolelli@psu.edu

ABSTRACT

Citation matching, or the automatic grouping of bibliographic references that refer to the same document, is a data management problem faced by automatic digital libraries for scientific literature such as CiteSeer and Google Scholar. Although several solutions have been offered for citation matching in large bibliographic databases, these solutions typically require expensive batch clustering operations that must be run offline. Large digital libraries containing citation information can reduce maintenance costs and provide new services through efficient online processing of citation data, resolving document citation relationships as new records become available. Additionally, information found in citations can be used to supplement document metadata, requiring the generation of a canonical citation record from merging variant citation subfields into a unified "best guess" from which to draw information. Citation information must be merged with other information sources in order to provide a complete document record. This paper outlines a system and algorithms for online citation matching and canonical metadata generation. A Bayesian framework is employed to build the ideal citation record for a document that carries the added advantages of fusing information from disparate sources and increasing system resilience to erroneous data.

Categories and Subject Descriptors

H.3.6 [**Information Systems**] Library Automation – *large text archives*.

I.2.3 [**Artificial Intelligence**] Deduction and Theorem Proving – *inference engines, uncertain, "fuzzy," and probabilistic reasoning*.

H.3.3 [**Information Systems**] Information Search and Retrieval – *clustering*.

General Terms

Algorithms, Management, Experimentation, Security

Keywords

Citation Matching, CiteSeer, Bayesian Inference

1. INTRODUCTION

Citations in research publications represent an important knowledge source regarding the context of scientific work. Since the introduction of the Science Citation Index [Garfield, 1964] citations have been used to measure research impact in terms of authors, publications, and publication venues. More recently, citations have been used to facilitate information search and retrieval in scientific digital libraries. Citation relationships have been shown to be valuable for tasks such as ranking search results, identification of related research documents, trend analysis, and social network analysis.

In collections of academic publications, citations represent relationships between documents. These relationships form a data structure generally known as a "citation graph", where documents are vertices and citations are directed edges between citing and cited documents. Constructing this graph requires discovering which documents are referenced by individual citations, a task that can be achieved through matching citation and document metadata. For each citation, the citation text must be parsed to find specific informational items such as authors, title, publication venue, publisher, editors, year of publication, and any other available information. The parsed metadata can then be used to find documents with the same or similar metadata. This process is complicated by frequent errors of information extraction as well as errors in the original citation. Additionally, stylistic variation results in identity uncertainty – for instance, citing a paper in "Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries" or simply "Proc JCDL" may both be acceptable formats.

Large citation indices such as ISI have historically depended upon manual information extraction, requiring human effort to tag and correct information in citations and to facilitate relationship discovery. This process is time-consuming and expensive such that citation indexing is typically beyond the capability of non-commercial digital research libraries. In recent years, work has shown that it is possible to handle the task of citation indexing automatically through methods of artificial intelligence. The CiteSeer Digital Library [Lawrence et al., 1999b] was created in 1997 to demonstrate autonomous citation indexing (ACI), and has since grown to a collection of over 725,000 documents with over 8 million citations. More recently, Google has released The Google Scholar[1], which incorporates ACI to index over 433 million document and citation records[2].

ACI represents a challenging automated data management task. The first step in this process is the extraction of citations from research papers and subsequent parsing of citation subfields to build accurate metadata for each citation. The problem of citation parsing remains unsolved and the best parsers to date, built using machine learning tools such as hidden Markov models and maximum entropy models, are error-prone and often produce noisy results. Errors at this level, along with errors in the citation text (such as typos), negatively influence subsequent development of the citation graph.

Finding the best document matches for citations is a problem of identity uncertainty [Pasula et al., 2003]. Citations are references to unique documents, but their representation can vary. The task of an ACI system is to uncover the identity of the paper that a citation refers to in order to group together citations to the same document, and to link citations to real documents – those that exist inside the ACI system and those that are yet uncollected. This is generally achieved by the batch clustering of citations into citation groups that are thought to represent a single document, and then matching the citation group to a real document. Clustering is typically done through string distance or similarity measures across citation subfields.

However, a fully automatic digital library is not only uncertain about the identity of citations, but it is uncertain about the identity of *the documents themselves*. Here, the identity of a document can be defined as the document's ideal metadata record. Even in situations where human-generated document metadata is available, the data can be incomplete or incorrect. This problem is amplified in digital libraries such as CiteSeer, in which the system is responsible for building its own document collection without the aid of human-produced metadata. Documents in CiteSeer enter the system with information extracted by automatic parsing algorithms, and the information is usually restricted to the title, author names, and any other metadata typically available in document headers. The parsed information is often noisy or incorrect, and almost always incomplete. In particular, information regarding the document's publication venue and year of publication are seldom captured. Citation information can be used to fill in missing information and to correct erroneous data. When citations to a document record are found, the document's metadata can be updated with information from citations. Information from multiple citations can be fused to form a "best guess" as to the correct metadata values for the document. This process will be referred to subsequently as the process of building the canonical metadata for a document.

The work presented here is part of a larger project to develop an improved version of CiteSeer, representing an exploratory study into novel approaches to fully automate citation management and determining document metadata. The proposed solution incorporates an on-line method for citation match resolution that updates the citation graph environment and resolves document identities "on-the-fly" in response to the ingest of new citations into the system. The solution promises to reduce the cost of system maintenance, provide more up-to-date information, provide confidence metrics and improved accuracy for document metadata, and has the added benefit of improving the integrity of information within CiteSeer databases.

1.1 Motivations

In [Petinot et al, 2004], a service-oriented architecture for a new CiteSeer was introduced. The architecture contained two sub-systems for citation management: a *citation extraction service* that extracts and parses citations from academic publications, and a *citation graph service* which maintains a citation graph representing the citation relationships between documents. The work presented here is an investigation into the design of the citation graph service, and is intended to serve as a framework for managing citation graph relationships and also as an inference engine for determining canonical document metadata for all document records. The goals in designing the citation management system are the following:

1) *Provide better document metadata*. Currently, CiteSeer builds canonical metadata by using the subfields of the most similar citation to a document from the document's citation group. This is unsatisfactory since there is no guarantee that the most similar citation contains the best metadata, or even that any citation contains the best representations of all metadata fields. This problem is addressed by the current work by using a data fusion approach based on Bayesian inference to combine data from citations into belief vectors in the values of document metadata. Each metadata element in the document record is supported by observations across all citations.

2) *Reduce the cost of maintenance*. Building the citation graph in CiteSeer is a time-intensive, offline task that takes several days to complete on low-tier enterprise hardware. The process must take place on a separate machine from the publicly accessible service and then the database must be synchronized after completion. Due to batch clustering, the addition of a single citation requires rebuilding the entire citation graph to include the new instance. The present work investigates the use of on-line citation matching such that the citation graph environment can be adjusted immediately based on a single new citation, eliminating the need for expensive batch updates, providing more up-to-date data, and eliminating the need for batch synchronization processes across servers.

3) *Allow the development of flexible APIs into CiteSeer's citation graph system*. On-line citation matching has the added benefit of providing increased access to the citation graph by users. For instance, users can gain an entry point into the graph based on a citation string that is passed to the system, parsed, and matched instantly to its citation graph context.

4) *Increase data resilience*. Although related to the goal of providing more accurate document metadata, this goal is focused on the generation of metadata from sources external

[1] http://scholar.google.com

[2] This number was discovered by searching for "+the" in the Google Scholar search engine.

to CiteSeer. CiteSeer provides an open, wiki-like approach to user-contributed metadata changes, known as distributed error correction [Lawrence et al., 1999a]. This framework allows users to change any document's metadata at will, through a web form. Mostly, user error correction is a great benefit; however, there have been several abuses of the system. Information obtained from other open archives can present the same problem in the case of egregious errors in document records. Metadata derived from the citation graph can be weighed against external information sources in order to assess the trustworthiness of a proposed change. For well-cited papers, multiple observations regarding the document's identity are readily available. The combined weight of these observations may be harnessed to create system alerts when information from an external source conflicts with previous knowledge.

5) *Improve citation matching performance*. Finally, a core weakness in CiteSeer's current graph maintenance system is addressed – that the process of building canonical metadata for a document cannot influence citation matches to that document in a principled, declarative manner. Once canonical metadata is determined for a document after a batch citation update, that metadata is fixed to the document record and cannot influence the citation clustering, even when citations that matched the old metadata no longer match the new metadata according to the original matching criteria. The present work provides a fluid framework for building canonical metadata in which all evidence for the metadata is always considered and easily fetched, and document metadata changes can have immediate impact on citation clustering.

1.2 Related Work

A standardized part of a scientific article, citations have great values to be explored, and there have been numerous works on parsing and analyzing citations for different purposes. Besides being used as objects in document retrieval and for establishing relationships between articles (e.g. hyperlinks in the context of the electronic documents on the Web), citations are also analyzed to provide quality assessments for documents and authors as citations are often assumed to represent the endorsement by the research community. Another example is [Katerattanakul et al., 2003], in which citations are used to assess the quality and impact ratings of the journals in the computer and information sciences domain, which proved effective. In [Lawrence et al, 2000], citations also served as an indication of the persistence of information on the Web. In recent years, citations are also used to identify relationships within a specific research community; typical works in this line include co-citation and co-authorship analysis. For example, [Egghe and Rousseau, 2002] used bibliographic coupling to discover subject similarity of scientific articles. [Hill and Provost, 2003] used citation matching to identify anonymous authors in double-blinded submissions.

CiteSeer represents the efforts to automatically discover, archive, and index online academic publications, and supports unique citation-based features, such as cross-document citation browsing, citation statistics, co-citation and citation graph analysis, etc. [Lawrence et al., 1999] presented four algorithms for citation matching, which were based on edit-distance, word matching, word and phrase matching, and subfield matching. Based on a comparison of the results, the word and phrase matching algorithm achieved the best accuracy while the subfield matching algorithm was the most efficient in computation. The current citation management sub-system in CiteSeer has not changed much since its initial debut at the NEC Research Institute. The Autonomous Citation Indexing (ACI) [Lawrence et al., 1999] employs heuristics based upon a number of invariants (aka. subfields that are relatively uniform in syntax, position, and composition) to parse the citation in a top-down fashion, and syntactic relationships between subfields to predict the existence of a desired subfield.

A number of studies have been conducted in order to enhance the performance in citation extraction and matching. [Bergmark, 2000] discussed the extraction of bibliography information from online literature. Here, reference parsing and matching were done using a number of linguistic cues observed from document samples. [Sarawagi et al., 2003] described a system to generate citation graphs as the authors' submission to the KDD Cup in 2003. Issues regarding bibliography information extraction and citation matching are discussed. [Besagni and Belaid, 2004] exploited the syntactic regularities in citation information, and used a method based on part-of-speech tagging to extract bibliographic elements in a bottom-up fashion.

The problem of identity uncertainty has been addressed using different methods. For example, [McCallum et al., 2000] proposed a two-stage clustering technique for identifying and matching identical bibliographic references. The citations are first divided into overlapping canopies and then further clustered through nearest-neighbor Greedy Agglomerative Clustering, the distance metrics of which are the edit-distances on the author, title, date, and venue fields. Name ambiguity being a special case of identity uncertainty, in [Han et al, 2004], the primary target was to disambiguate variants in author identities. Two techniques based on a naïve Bayes model and an SVM were presented. Similar approaches can be used to disambiguate variants in title and publication venue strings as well.

[Pasula et al, 2003] and [Marthi et al., 2003] used a probabilistic model for citation extraction. Similarly, [Wellner et al., 2004] recommended an iterative approach to citation extraction and matching, in which the uncertainty about citation fields is modeled to enhance the accuracy of co-reference, and in turn the co-reference information can be used to improve the accuracy of extraction. [Culotta and McCallum, 2004] presented a way to quantitatively measure the confidence in information extraction tasks. Applying this technique to citation extraction and clustering tasks, the confidence values attached to the citation tags can be examined so that alternative labels may be considered for higher extraction and clustering accuracy.

1.3 Organization

The remainder of this paper is organized as follows. Section 2 describes a basic index method for identifying matches between document and citation records. Section 3 presents a method using Bayesian inference to identify canonical document metadata based on citation metadata and other information sources. Section 4 brings together the previous methods with an iterative approach to determining citation relationships and canonical metadata. An implication of proposed methods for data security is discussed in Section 5 and Section 6 presents an evaluation of the clustering and metadata determination methods. Section 7 presents conclusions and directions for future investigation.

2. MATCHING CITATIONS AND DOCUMENTS

In CiteSeer, linkages between documents are represented by citation relationships. However, the citation graph is currently maintained by an off-line batch program, which is difficult to control and error-prone. In addition, citations and document records are treated differently in the current CiteSeer system, ignoring the fact that the cited objects are themselves documents. This complicates the system unnecessarily. To simplify the model, the concept of a "virtual" document record is defined in the system. Once a new citation is ingested into the repository, the system searches the existing repository in order to identify the referenced document. If a match is found, an edge is added to the citation graph. Otherwise, the system creates a virtual document for the cited document, which will take on the extracted metadata of the citation. The record is called "virtual" because the cited document is not in the repository. If the real document enters the system on a subsequent update, the document metadata is used to search the repository and in the best case it is matched to the corresponding virtual record. The virtual record is then updated with a pointer to the document file, making it a "real" document record. This new framework makes the CiteSeer repository more unified and complete. There are no citation edges pointing to an external unknown resource. All edges are internal in the document database and "real" as well as "virtual" documents can be searched in the same index space.

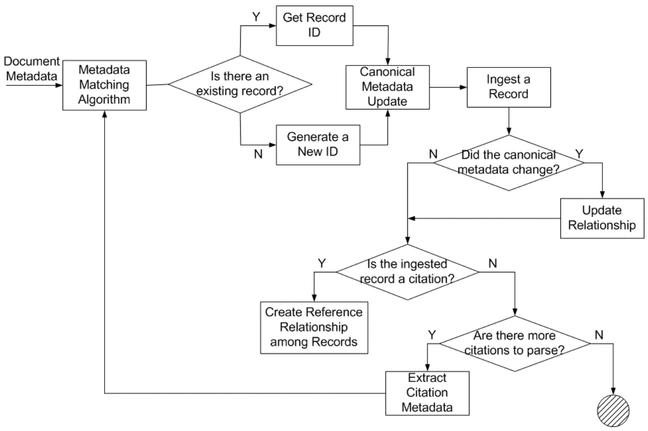

Document Metadata
Metadata Matching Algorithm
Is there an existing record?
Y
Get Record ID
N
Generate a New ID
Canonical Metadata Update
Ingest a Record
Did the canonical metadata change?
N
Y
Update Relationship
Is the ingested record a citation?
Y
N
Create Reference Relationship among Records
Are there more citations to parse?
Y
N
Extract Citation Metadata

Figure 1. Paper Ingestion and Update Workflow.

The core module of this online process is the clustering algorithm, matching citations to documents and matching documents to citations. This algorithm makes use of full-text indices, which are built on citation metadata and document metadata respectively, to locate the relevant records. In realistic cases, there are often abbreviations and typos in the citation metadata. Our method is designed to operate despite some data inaccuracy. In our implementation we utilize the open source Lucene [Apache Lucene, 2005] index framework, harnessing its native fuzzy search capabilities. For example, to retrieve the matching documents to a citation, the algorithm performs the following steps:

1. Relevant metadata fields (authors, title, etc.) are extracted from the input texts.
2. A fuzzy query is formulated, using the querying terms obtained in Step 1 and a pre-defined similarity threshold.

   The metadata elements considered for the match are limited to title and author fields for this initial study. According to our observations of citation records in CiteSeer, typos and abbreviations often happen in the author attribute, while the title attribute usually contains fewer variations. Thus, the fuzzy queries are constructed with the following rules:

   For all tokens taken from the author attribute $\{a_1, a_2, \ldots, a_n\}$, conjunction operators are used to connect the query terms: $a_1 \vee a_2 \vee \cdots \vee a_n$. For tokens in the title attribute $\{t_1, t_2, \ldots, t_m\}$, disconjunction operators are applied: $t_1 \wedge t_2 \wedge \cdots \wedge t_m$. Thus, the final query clause can be represented as:

   $$Q = ((a_1 \vee a_2 \vee ... \vee a_n) \wedge (t_1 \wedge t_2 \wedge ... \wedge t_m)) \sim \alpha ,$$

   where α is the similarity threshold in a fuzzy query.
3. The query is issued to the corresponding index and a series of possibly relevant records are returned by the search engine.

In this algorithm, the operation interfaces are provided as follows:

1. findDocMatches C, returns a list of documents surpassing a given similarity threshold to C.
2. findCitMatches D, returns a list of citations surpassing a given similarity threshold to D.
3. getSimilarity C, D, returns the similarity of citation C and document D.

In the above operation definitions, C is a citation metadata record, while D is a document metadata record.

This online matching algorithm, along with the metadata cluster repair algorithm discussed in the subsequent sections, can effectively update the citation graph progressively, without a batch-mode post-treatment. Thus, the data integrity and consistency can be guaranteed at any time and the entire process is autonomous.

3. LEARNING FROM OBSERVATIONS

Although the index-based citation matching framework provides a means to cluster citations to documents, discovering citation relationships based on existing metadata is only the first stage of the proposed scheme for citation clustering. The next task is to derive canonical metadata for each document record R based on the metadata of all citations linked to R. This process can be augmented by any other information source for R.

This work casts the problem of metadata identification as the problem of generating beliefs in the identity of a document based on observational evidence, where a document's identity is represented by its ideal metadata record. In an autonomous digital library such as CiteSeer, documents enter the library's world tagged with incomplete and often incorrect metadata. Our confidence in what metadata is initially available depends on the accuracy of the parsing methods employed. From the time of initial entry into the database, records may be linked with other information sources – citations are the most prevalent of such sources, but documents may also be linked to external metadata records with more accurate and complete data (for example, records linked from the DBLP), and document records may be corrected by users.

Each information source linked to a document can be viewed as evidence of the document's ideal metadata record. Through multiple observations of linked sources, it is possible to aggregate this evidence in order to form beliefs about the "true" identity of documents. Bayesian inference provides a natural framework for representing beliefs based on evidence, and will be used to create a belief engine for all metadata elements of each document. The prototype system described here considers only author names, titles, and publication years.

3.1 Network Structure and Belief Updates

Each evidence-driven metadata element X in a record R has associated with it a Bayesian belief network $B_{R(X)}$, responsible for deciding the canonical value X′ from all observations on X. The purpose of each network $B_{R(X)}$ is to develop degrees of belief in each possible value X, and X′ is chosen based on the value with the largest belief score. The network structure in this prototype system is quite simple, using a two-layer tree to represent an arbitrary number of independent observations on a single target node (see Figure 2). In this framework, node values are represented as vectors indexed on all possible outcomes.

Node X is represented by the probabilities π(x) across all possible x. The values of π(x) are determined by the standard equation

$$\pi(x) = P(x \mid o_x)$$

where o_x represents all observations on X. In this simple scheme, the belief vector BEL(x) is simply π(x) since x is determined solely based on the predictive support of the observations o_x. Given a prior belief vector BEL(x), BEL(x) can be updated with a new observation o_x' using only a local computation. From [Pearl, 1988], when a new observation O is received for X, O sends a message to X with the new observation vector o′(x) and the new belief vector BEL(x) can be computed by the equation

$$BEL(x) = \alpha\pi(x)o'(x) \quad \textbf{(1)}$$

where α is a normalizing constant to ensure that ΣBEL(x) = 1.

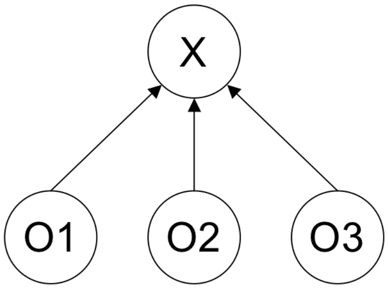


Figure 2. A simple Bayesian network where three independent observations affect the belief in the value of X′.

Observations will enter the system as metadata records, and must be translated into vector representations. Since each observation only has knowledge of its own metadata, a lookup table L_x must be used to generate a vector index position for the observed value. An example observation vector o′(x) may be (0, 0, 1, 0), indicating that $L_x(2)$ is the observed value for x. This vector must then be adjusted based on our confidence in the observation. This is achieved using a confidence matrix

$$M_{x|o} = P(x \mid o) = \begin{pmatrix} C \; if \;\; x = o \quad x,o = 1,2,...,n \\ (1-C)/n-1 \; if \;\; x \neq o \quad x,o = 1,2,...,n \end{pmatrix} \quad \textbf{(2)}$$

where C is an arbitrary confidence score ascribed to o, and n is the number of possible values for x. The message sent to X by an observation is then determined by the value $o'(x)M_{x|o}$. This has the effect of reducing the confidence in the value predicted by o′ and sharing the uncertainty over the other possible values. Taking the initial example of o′(x)=(0,0,1,0), assigning C=0.7 to o′ results in an actual message of (0.1,0.1,0.7,0.1) sent to X. Although it is not necessary that observation vectors sum to 1, this confidence scheme captures the uncertainty in observations in the desired manner. In practice, the confidence metric C can be determined by measuring the accuracy of data sources. Observations from document parses receive C=F, where F is the F-score of the parsing algorithm used. Confidence in citations and external metadata records must be measured by multiplying the source accuracy with the matching accuracy, representing the propagation of errors at various levels before an observation is matched to X. Observations based on user corrections are a special case, discussed in a later section.

Each metadata element is determined by its own belief network, such that no single data source has a monopoly on all metadata fields. This provides a simple scheme for determining single-valued metadata fields such as title and year; however, multi-valued fields with order properties deserve special attention. Multiple metadata elements may be required for multi-authored papers, and it is desirable to model belief not only in the value of a specific author name, but also the value's position in the author list. This is handled simply by creating a separate network for each observed author position. Since the correct number of authors must be preserved, it is necessary to create a NULL output value for each author position. This way, if an observation O_1 provides evidence for three author positions and a separate observation O_2 provides evidence for only two, O_2 can also be used as evidence that the value of the third author is NULL.

4. CLUSTER REPAIR

Since a document's citations are identified based on text similarity to the document's metadata, adjusting metadata dynamically in response to new evidence can lead to inconsistencies in citation groups. Repairing citation clusters can lead to better clustering accuracy as well as increased confidence in metadata values. To handle this task, the following recursive algorithm is proposed, to be called every time a metadata change occurs in a record:

repairCluster(R)

```
Find matching citations M for R
For each citation C in $G_R$
  If C is not contained in M
    Add C to REVOKE
  Set $G_R$ = M
  Reset belief vectors
  For each citation C in $G_R$
    If C is not contained in REVOKE
      Update belief vectors using C
  If metadata changes
    repairCluster(R)
```

Definitions:

R: A document metadata record.
G_R: the citation group of R.
REVOKE: an array of citations that have lost voting privileges.

In line 1, the citation matching approach described in Section 2 is employed to find all citations matching R's new metadata. Lines 2-4 remove citations that previously matched, but no longer match R's metadata. Removed citations temporarily lose their privileges to affect R's metadata for the duration of the repairCluster call stack. The reasons for this are discussed below. Line 5 sets R's citation group to the newly matched citations. Line 6 regenerates the belief vectors for all elements of R's metadata, based only on observations other than citations, and lines 7-9 rebuild the belief vectors based on all privileged citations in the new citation group. The performance of the repairCluster algorithm is affected by two heuristics: the concepts of shared citations and voting privilege.

Shared Citations

A single citation can be linked to multiple documents. If the similarity between a citation and a document passes a given threshold, the relationship is determined to be viable no matter how many other documents also surpass the similarity criterion.

This is helpful in two ways:

- **Representational honesty.** If there exists a significant degree of uncertainty as to which document in a set of documents a citation belongs to, it is desirable to link the information to all possible targets in order to explicitly express the uncertainty to system users, and to the system itself. These shared citations can be easily tagged in order to link all target documents to a citation through system APIs and the user interface.
- **Prevention of update cascades.** If relationships are only binary, updating Document A can result in citations being "stolen away" from Documents B and C. In order to maintain metadata and graph integrity, it would then be necessary to update Documents B and C, which could steal citations from other documents, and so on. The one-to-many relationship model provided by shared citations prevents these cascades.

Privilege Heuristic

Cyclical operation of the algorithm, where citations are removed and then added back at a lower level of the repairCluster call stack, could result in unbounded iterations. It is conceivable that one citation C1 could be removed, causing a metadata change that results in the inclusion of citation C2, causing a metadata change resulting in the addition of C1, causing the removal of C2, and so on in an infinite loop. To prevent this, the "voting privilege" is revoked from any citation removed from the citation group during a single call stack. More precisely, once a citation C1 is removed from G_R, it can return to G_R but it cannot influence metadata belief vectors for the remainder of the repairCluster iterations. This heuristic is reasonable if a progressive tendency toward correct metadata is observed, and it is not desirable to permanently ban citations from influencing a document's metadata. At the end of a repairCluster call stack, the non-voting citations regain voting privileges. Although this heuristic can result in inconsistent citation groups, the inconsistency is traded for proven convergence. In practice, this heuristic has not been needed; however, it will be incorporated until proven unnecessary.

Only the addition or removal of a citation with voting privileges can result in a metadata change. Since a removed citation forfeits voting privileges in a call stack, only N - size(G_R) citations can be added with voting privileges and only N citations with voting privileges can be removed, where N is the number citations in the database and size(G_R) is the size of the R's citation group at the first call in a stack of calls to repairCluster. Therefore, the upper bound on the number of possible iterations in a single call to repairCluster cannot exceed N+N-size(G_R), or 2N. In practice, very few iterations have been needed, as discussed in Section 6.

4.1 Garbage Collection

The cluster repair algorithm can be extended slightly in order to facilitate garbage collection in the document database. Here, garbage collection refers to the process of expunging virtual document records (described in Section 2) that are no longer useful. Virtual documents are created in order to provide a match for citations that do not match any "real" document in the database. They may be transformed into real document records when a document enters the system with matching metadata; however, virtual documents may also develop metadata that does not match any existing document. In this case, virtual documents will persist indefinitely until explicitly deleted. A simple reference count method can be adopted for garbage collection on virtual document records. In this model, information sources linked to a document record act as references. Due to shared information sources, two types of references can be identified:

1) A "strong" reference is any single information source that is linked to only one document record. For example, a citation that matches only one document record is a strong reference. Likewise, a metadata record from an external source (such as the DBLP) that matches only a single document record is a strong reference.

2) A "weak" reference is any single information source that matches multiple document records. Cases of weak references include any shared citation or shared support from an external metadata record.

In the proposed garbage collection scheme, a virtual document must have at least one strong reference to avoid collection. This gives preference to real document records and "better" virtual document records. Once a virtual document is deleted, its information sources may become strong references to other linked records as long as the sources are no longer shared. Whenever any information source is linked to a record in calls to repairCluster or when new documents enter the system, a signal is sent to check all virtual records linked to the information source. The virtual records are examined to determine support from strong reference; if no strong references are identified the record is deleted.

5. DATA INTEGRITY

As mentioned previously, document metadata records in CiteSeer may be changed by users at will. Obviously, a major concern for publicly modifiable data resources is protection against malicious user updates. This is primarily a concern for wiki web pages, but it is a concern for CiteSeer as well due to the wiki-like distributed error correction model. In practice, the vast majority of user updates in CiteSeer result in valuable corrections to parse errors or the inclusion of previously missing data; however, there have been a few rare exceptions where the error correction mechanism was used to provide garbage information for controversial papers and to post spam advertisements on highly cited papers. Currently, human observation and intervention is necessary to identify and correct these abuses. Although the vulnerability presented by distributed error correction is serious, the value of user corrections has outweighed the risk to date. CiteSeer logs as many as 30 corrections per day. Like wiki pages, history information is generally available for CiteSeer to role back to a previous metadata version if a problem is discovered. However, it is better for a system administrator to detect malicious updates rather than a user for a variety of reasons. Unfortunately, for a large, well-used library, odds favor user discovery of record errors unless a proactive approach to erroneous update detection is employed.

The evidence-driven metadata determination framework proposed in Section 3 provides a means to automatically determine when potentially malicious or otherwise incorrect data updates are provided. A user correction to a document record R can be treated as another observation regarding R's identity, although a special kind of observation. An empirically derived confidence score C is assigned to all user corrections and corrected metadata is then compared to the existing belief vectors in R. For example, consider R has a prior belief vector

$$\pi(R(E)) = (0.8, 0.05, 0.1, 0.05)$$

in metadata element E. This represents the belief that the value represented by $\pi(R(E))(0)$ is the most likely value for E. A user correction vector

$$O_{R(E)} = (0,1,0,0)$$

is submitted to R with an assigned confidence C=0.9. The belief vector BEL(R(E)) is updated by Equations 1 and 2 with the user supplied vector $O_{R(E)}$ resulting in a new belief vector

$$BEL(R(E)) = (0.346, 0.589, 0.043, 0.022).$$

The combined evidence supports the user's interpretation for the value of E, that is, the second element in the vector is now the most likely value. If, however, we have higher prior confidence in a value of E, the results can be different. Consider the prior belief vector

$$\pi(R(E)) = (0.9, 0.025, 0.05, 0.025).$$

The result of adding $O_{R(E)}$ is then

$$BEL(R(E)) = (0.543, 0.412, 0.03, 0.015).$$

From the result vector, we can see that the combined evidence supporting our belief in the value of E is at odds with the value supplied by the user. In this event, the correction can be logged but not applied, and the system can notify an administrator of the conflict. The determination of the proper confidence to assign to user corrections is left to future work.

If a user correction is accepted, the corrected data should then be locked to prevent further updates from the belief network B_R. Imagine the frustration of an author who meticulously corrects parse errors in an important publication only to see those corrections "updated" after a few days. The lock can be achieved by assigning a belief score of 1 to the corrected metadata element values and 0 to all other possible values. No amount of belief updating can surmount this level of certainty.

There are two approaches for dealing with locked metadata elements in the database – what can be termed the "trusting" approach and the "skeptical" approach. In the trusting approach users are trusted, within the bounds of the proposed malicious update detection system, to provide accurate metadata that should never be changed. The belief vector for the updated document record R is modified to represent certainty and future observations by the system to R's identity cannot result in metadata changes.

In the skeptical approach, the user correction is only tentatively accepted. The metadata of the corrected document is split into two records representing potentially disparate metadata. In this scheme, the user-provided metadata is taken as the primary record for a document and will be presented to system users; however, the system also maintains a "shadow" record based on system observations alone. Thus, the shadow record represents the state of a document's metadata as if a user correction had never occurred. Each record maintains its own citation cluster. The incorporation of a shadow record can be useful for determining malicious or incorrect user updates in the future. For example, if a user posts incorrect data to a document with very few citations (perhaps the document was recently published), the confidence assigned to the user's correction may outweigh all other collected evidence. In this case, the malicious update is allowed. In the future, enough citations may be collected for the shadow record that the weight of the new evidence can challenge the user-supplied data, notifying an administrator when a conflict is detected.

6. EVALUATION

This section discusses the performance of the described methods on a test bed of citation and document records. A citation data set was created from the existing CiteSeer document and citation repositories. Ten frequently referenced document records were selected from the top of CiteSeer's most-cited document list along with all corresponding citations. The citations were manually reviewed and only correct citation matches were kept. 9,121 citations were used in the final test set. The citations in the data set represent a biased sample, having been identified as matches to the related documents by CiteSeer's internal algorithms. In order to induce random variability into the citations and better test how the algorithms cope with possible typos and coding errors in the attribute values, the data set was run through a noise generation program to purposely add some noise into the citation records. Seven categories of noise are created by the generator:

- Randomly insert a word into the title.
- Randomly delete a word from the title.
- Randomly insert an author name.
- Randomly delete an author name.
- Randomly misspell a word in the title.
- Randomly misspell an author name.
- Mistakes in the publication year attribute.

Corresponding parameters are provided to control the probability with which a certain category of noise will occur, varying from 0 to 1. A noise rate of 0 means the original version of citation texts are adopted, without any intended modifications. A noise rate of 1 means a type of noise is destined to happen.

6.1 Index-Based Citation Clustering

We want to explore how well our new document matching algorithm works under different working contexts, using the convention of precision and recall. In the experiments, only the author and title attributes are used for determining citation matches. An ideal citation graph is pre-extracted and encoded in the citation tag. All the 9,121 citations are used as input to query against the document index. The returned correspondence lists are compared with the ideal citation graph to identify the correct links.

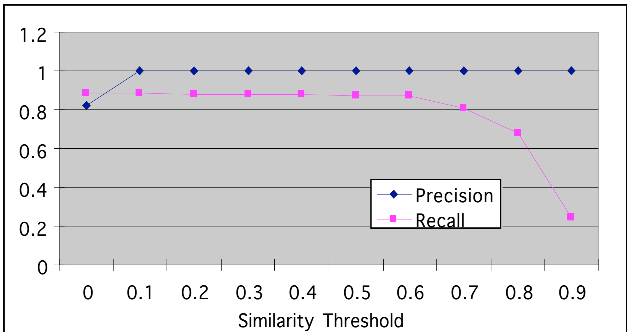


Figure 3. Studying the effects of similarity threshold to precision and recall

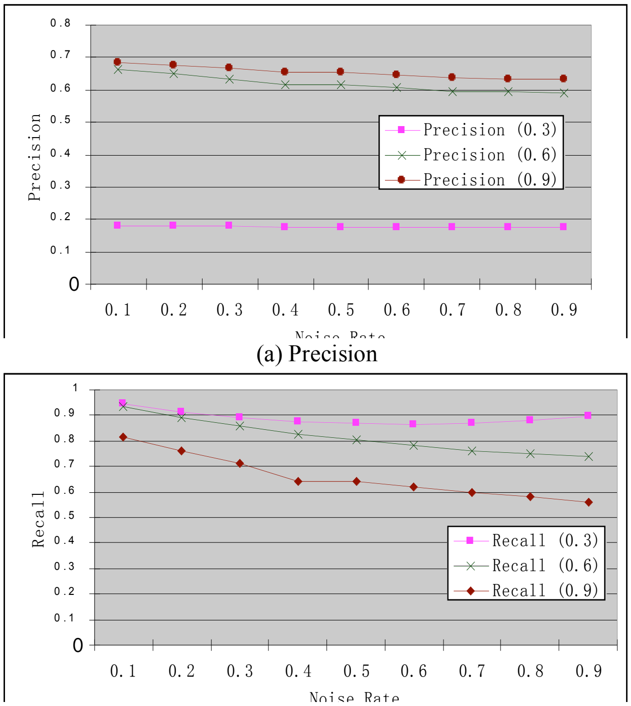
In the first experiment, we take the original citation set as the input dataset, varying the similarity threshold from 0 to 0.9 to observe the influence of similarity threshold on the returned precision and recall. When a lower threshold is bound to the fuzzy query, a higher recall can be obtained, while the precision decreases. In contrary, a high threshold can strongly limit the scope of returned matches, resulting in high precision and low recall. The corresponding relationship is shown in Figure 3. The precision remains very high even with a low threshold (around 0.3) because of the usage of disconjunction operators in titles. As well, the recall decreases very slowly as the similarity threshold increases until 0.8, higher than that of the clustering algorithm implemented in the current CiteSeer (around 0.7). The results show that if a reasonable similarity threshold is defined, the new matching algorithm can provide both a good precision and a good recall.

Next, noise is introduced into the citation data to test the capability of the matching algorithm to handle inaccurate inputs. We define a uniform noise rate for the citation, which sets all the noise parameters discussed in the previous section of the noise generator to be the same. The noise rate varies from 0.1 to 0.9. In addition, we also vary the similarity threshold (0.3, 0.6 and 0.9) to compare the precision and recall curves. In addition, the query clause is reformatted to cope with an error-prone dataset. Conjunction operators take the place of disconjunction operators among the title terms as well, making the query syntax to be

$$Q' = ((a_1 \vee a_2 \vee ... \vee a_n) \wedge (t_1 \vee t_2 \vee ... \vee t_m)) \sim \alpha \cdot$$

The results are shown in Figure 4.

In Figure 4(a), all precision curves drop gradually as the noise rate grows up. Because conjunction operators are chosen to connect terms in titles, the precision is low for a lower similarity threshold (0.3). However, the precision varies between 0.6 and 0.7 for a threshold that is not too small. In Figure 4(b), the recall curves for all the three thresholds share a similar shape, decreasing slowly. If a median threshold is chosen, say 0.6, the recall value remains higher than 0.7 even more noise is added to the original citation texts. Based on the above observations, a well-chosen threshold can achieve both good precision and good recall in spite of possible noises.



(a) Precision

(b) Recall

Figure 4. Comparing the effects of noises in different similarity thresholds

6.2 Metadata Determination and Cluster Repair

In the evaluation of the index-based cluster algorithm, document records with perfect metadata were used to test matching performance. In this section, the system is evaluated with noisy document records, using the document metadata determination and cluster repair methods discussed in Sections 3 and 4. Document records were subjected to the same noise-inducement process as the citations in section 6.2, except that each document record was subjected to each type of corruption, guaranteeing that each document record was noisy. Authors were not deleted from single-authored papers, although the single author name could become corrupted. Document records were used to query the citation database using the method described in Section 2. Based on the citation results, the metadata of the querying document was updated per the methods of Section 3. Confidence in the document metadata was arbitrarily set at 0.8, and confidence in citation data was set at 0.5. Since the vast majority of information sources in the test data are citations, the confidence scores are relatively unimportant to the results. The cluster repair algorithm of Section 4 was then used to iteratively query the citation index and repair the document's metadata until convergence. Only title, author, and year metadata was tested for accuracy.

The goal of determining canonical metadata is to uncover the ideal metadata for each record. If the ideal metadata is achieved, clustering performance is the same as reported in Section 6.2 at all citation noise rates. The canonical metadata determination results were measured in terms of the percent of metadata elements in all documents that were determined correctly. This includes all author positions as individual elements, titles, and publication years of the documents. The results are presented in Figure 5. Perfect accuracy was achieved even at very high noise rates, only dropping off at 0.7 noise.

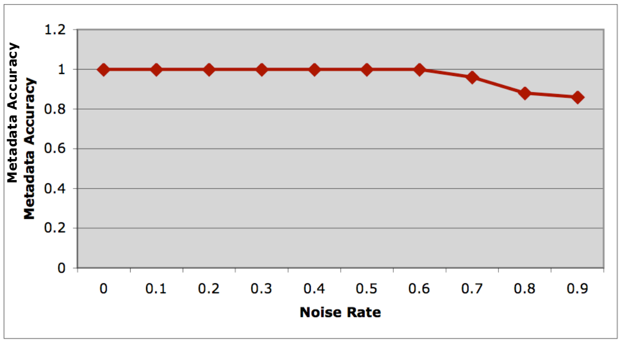


Figure 5. Metadata accuracy by noise rate.

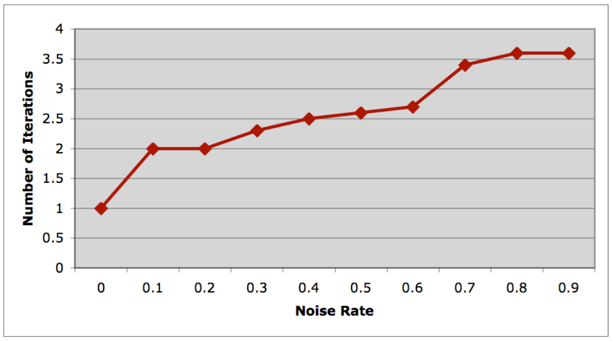


Figure 6. Number of cluster repair iterations required to converge citation clusters at various noise rates.

One concern regarding the cluster repair algorithm is performance, particularly regarding the number of iterations required to converge the cluster. One index call is required per iteration, and each call is computationally expensive. The theoretical bound on the number of iterations is 2N, where N is the number of citations in the database. The worst-case performance would be impractical in a large citation database such as CiteSeer, requiring millions of index calls to resolve a citation update. However, experimental results are encouraging and no call to the cluster repair procedure required more than four iterations to converge on the test data. The average number of iterations required to converge document citation groups at all noise rates is presented in Figure 6. At noise rate 0, only one cycle was required since linked citations never resulted in a metadata change for the already perfect document record. At rate 0.1 and 0.2, exactly 2 iterations were needed since each document record contained inaccuracies, and perfect metadata was achieved on the first iteration. As the noise rate was increased, the average number of iterations slowly increased, and the number of iterations jumped upwards starting at noise 0.7, just as metadata determination ceased being perfect.

7. CONCLUSIONS AND FUTURE WORK

This work provides a framework for clustering citations, building a citation graph, and determining canonical document metadata "on the fly." Citations are matched individually to documents, and citation groups arise implicitly as all citations linked to the same document records. Citation matching is achieved using Lucene, an advanced open-source indexing framework. Canonical metadata for document records is determined using simple Bayesian inference across all citations linked to a document, as well as other supporting information sources. The probabilistic framework for metadata determination also provides a method for protecting database records against malicious updates by users. This is particularly useful for open systems such as CiteSeer that depend on user-provided data for error corrections. Finally a method was described for repairing citation clusters dynamically in response to metadata updates.

The proposed scheme for citation matching has been shown to perform acceptably well on a noisy test set. There is room for improvement in the results, and future experiments will attempt to increase the precision of results by increasing the match similarity threshold according to system confidence in document metadata. The method for determining canonical metadata is quite robust to noise, and can be used to determine 100% accurate metadata for document records based on considerably noisy evidence. Although only document title, author names, and publication years were considered in this study, work is underway to extend the system to all metadata fields.

Performance and scalability issues of the system will be investigated further by creating a large-scale prototype to manage the citation graph for the entire CiteSeer collection. System performance will be assessed in terms of the speed of data updates and computational resources required. The accuracy of the resulting large-scale citation graph and document metadata can be assessed through a comparison with CiteSeer's current graph and metadata through manual review of randomly sampled document records from each system.

Two heuristics described in Section 4 lead to theoretically unsatisfying properties in the proposed approach to cluster repair. Citations may be shared among multiple documents, but the uncertainty in the correct document match is not represented when choosing metadata for each document. This property has the benefit of requiring only local computations on a single document node to repair clusters but leads to unintuitive reasoning about document metadata. In addition, the voting privilege heuristic used to prevent cycles in citation cluster repair procedures is unsatisfying since it can lead to inconsistency in citation clusters. It remains unknown whether this heuristic is needed, since it has not been required in any of the conducted experiments. Future work will investigate the effects of relaxing each of these restrictions, and more theoretically satisfying solutions to cluster repair convergence will be sought.

8. ACKNOWLEDGEMENTS

This work was sponsored by NSF, NASA, and Microsoft Research.

9. REFERENCES

[1] Apache Lucene. Apache Software Foundation, http://lucene.apache.org/java/docs/index.html, 2005.

[2] Bergmark, D. Automatic extraction of reference linking information from online documents. Technical Report 2000-1821, Computer Science Department, Cornell University, 2000.

[3] Besagni, D. and Belaid, A. Citation Recognition for Scientific Publications in Digital Libraries. *The 1st International Workshop on Document Image Analysis for Libraries (DIAL'04)*, Palo Alto, California, 2004.

[4] Culotta, A. and McCallum, A. Confidence Estimation for Information Extraction. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2004.

[5] Egghe, L. and Rousseau, R. Co-citation, bibliographic coupling and a characterization of lattice citation networks. *Scientometrics*, 55, Number 3, pp. 349–361, 2002.

[6] Garfield, E. Quantitative measures of communication in science. *Science* 144, pp. 649-654, 1964.

[7] Giles, C. L., Bollacker, K., Lawrence, S. CiteSeer: An Automatic Citation Indexing System. In *Proceedings of the 3rd ACM Conference on Digital Libraries*, pp. 89–98, 1998.

[8] Hill, S. Provost, F. The myth of the double-blind review: author identification using only citations. *ACM SIGKDD Explorations*, 5, Number 2, pp. 179 – 184, 2003

[9] Lawrence, S., Bollacker, K., and Giles, C. L. Distributed Error Correction. In *ICDL*, Berkeley, CA, USA, p. 232, 1999.

[10] Lawrence, S., Coetzee, F., Glover, E., Flake, E., Pennock, D., Krovetz, B., Nielsen, Finn, Kruger, A., Giles, C. L. (2000). Persistence of Information on the Web: Analyzing citations contained in research articles. In *Proceedings of CIKM 2000*, McLean, VA, USA, pp. 235–242, 2000.

[11] Lawrence, S., Giles, C. L., and Bollacker, K. Digital Libraries and Autonomous Citation Indexing. *IEEE Computer*, 32, Number 6, pp. 67–71, 1999.

[12] Marthi, B., Milch, B., and Russell, S. First-Order Probabilistic Models for Information Extraction. *IJCAI 2003 Workshop on Learning Statistical Models from Relational Data*, Acapulco, Mexico, August 2003.

[13] McCallum, A., Nigam, K., and Ungar, L.H. Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching. In *Proc KDD*, Boston, MA, 2000.

[14] Pasula, H., Marthi, B., Milch, B., and Russell, S., and Shpitser, I. Identity uncertainty and citation matching. *Advances in Neural Information Processing (NIPS)*, 2003.

[15] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Inference*. San Francisco: Morgan-Kaufmann Publishers, Inc., 1988.

[16] Petinot, Y., Giles, C. L., Bhatnagar, V., Teregowda, P. B., Han, H., and Councill, I. A Service-Oriented Architecture for Digital Libraries. In *Proc International Conference on Service Oriented Computing*, 2004.

[17] Sarawagi, S., Vydiswaran, V., Srinivasan, S., and Bhudhia, K. Resolving citations in a paper repository. In *Proc SIGKDD,* 5, Number 2, pp. 156-157, 2003.

[18] Katerattanakul, P., Han, B., Hong, S. Objective quality ranking of computing journals. *Communications of the ACM*, 46, Number 10, pp. 111 - 114, 2003.

[19] Wellner, B., McCallum, A., Peng, F. Hay, M. An integrated, conditional model of information extraction and coreference with application to citation matching. *In Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pp. 593-601, 2004.