Scholarly Big Data Information Extraction and Integration in the CiteSeer$^{\chi}$ Digital Library

Kyle Williams[‡,1], Jian Wu[‡,2], Sagnik Ray Choudhury[‡,3], Madian Khabsa[†,4], C. Lee Giles[†,‡,5]

[‡] *Information Sciences and Technology,* [†] *Computer Science and Engineering*
Pennsylvania State University, University Park, PA 16802, USA
[1] kwilliams@psu.edu, [2] jxw394@ist.psu.edu, [3] szr163@ist.psu.edu, [4] madian@psu.edu, [5] giles@ist.psu.edu

***Abstract*—CiteSeer$^{\chi}$ is a digital library that contains approximately 3.5 million scholarly documents and receives between 2 and 4 million requests per day. In addition to making documents available via a public Website, the data is also used to facilitate research in areas like citation analysis, co-author network analysis, scalability evaluation and information extraction. The papers in CiteSeer$^{\chi}$ are gathered from the Web by means of continuous automatic focused crawling and go through a series of automatic processing steps as part of the ingestion process. Given the size of the collection, the fact that it is constantly expanding, and the multiple ways in which it is used both by the public to access scholarly documents and for research, there are several big data challenges. In this paper, we provide a case study description of how we address these challenges when it comes to information extraction, data integration and entity linking in CiteSeer$^{\chi}$. We describe how we: aggregate data from multiple sources on the Web; store and manage data; process data as part of an automatic ingestion pipeline that includes automatic metadata and information extraction; perform document and citation clustering; perform entity linking and name disambiguation; and make our data and source code available to enable research and collaboration.**

I. Introduction

Scholarly big data refers to the vast quantity of data that is related to scholarly undertaking, such as journal articles, conference proceedings, theses, books, patents, presentation slides and experimental data. Big data is often described by a set of *V's*, which originally referred to volume, velocity and variety [18] but has since gone on to include other concepts such as value, veracity, viscosity and vulnerability. As evidence of the volume and velocity of scholarly big data, Microsoft Academic is reported to have over 50 million academic document records and in 2010 it was estimated that the annual growth rates of several popular databases from 1997-2006 ranged from 2.7 to 13.5% [19]. Furthermore, a large proportion of that data is freely available on the Web with a recent study finding that an average of 43% of articles published between 2008 and 2011 were freely available online [1]. The variety of big scholarly data is evident from the wide variety of examples given above, such as articles and lecture slides. Lastly, this data is of significant interest to groups involved in decision making in funding, education and government, as well as scientists, businesses and the general public. Given the scale of scholarly big data as well as the interest in accessing and making use of it, a number of services have arisen that collect, analyze and provide access to this data, such as Google Scholar[1], PubMed[2], the ArXiv[3] and CiteSeer$^{\chi}$[4].

A significant challenge for these services is to deal with the scale of the data they collect, integrating information from multiple sources and extracting meaningful information from the data. Some services, such as the ArXiv, allow people to submit documents and supply metadata for those documents while others, such as CiteSeer$^{\chi}$ automatically collect documents from the Web and perform automatic information extraction. One of the benefits of this automated process is that it allows for better scalability in terms of collecting and processing new scholarly data; however, it comes at the cost that the quality of the data may not be as good as that of the manually curated data due to the heterogeneous nature of scholarly data on the Web and the difficulties that arise in automatically extracting information from such data. Thus a challenge lies in designing algorithms and processes that are able to deal with this heterogeneity.

Scholarly data is also highly relational: citations among papers result in a rich citation network; co-authorship results in a co-authorship network; research projects are related to specific grants; and authors are related to specific institutions and publications. Furthermore, due to the heterogeneous nature of scholarly big data on the Web, multiple variations of named entities may appear. For instance, *NSF* and *National Science Foundation* may both refer to the *National Science Foundation* or the former may refer to the *National Sanitation Foundation*. Similarly, *C. Giles* and *C. Lee Giles* may or may not refer to the same author. Thus, a challenge lies in entity linking and name disambiguation in scholarly big data.

Another challenge still lies in sharing data. For instance, there are issues related to intellectual property and copyright that may limit the copying and sharing of data among different groups. Furthermore, the size of the data may also be a prohibiting factor.

In this paper, we describe our approach in addressing

[1] http://scholar.google.com/
[2] http://www.ncbi.nlm.nih.gov/pubmed/
[3] http://arxiv.org/
[4] http://citeseerx.ist.psu.edu/

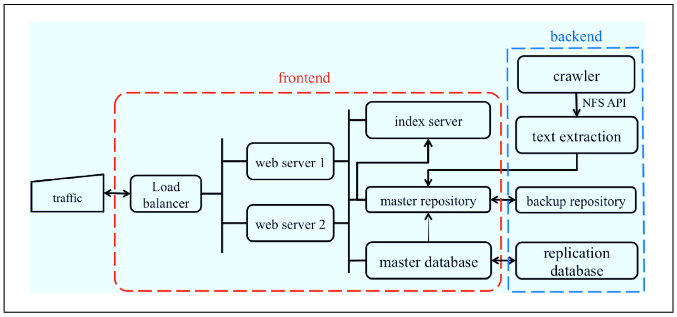

Fig. 1. CiteSeer^X architecture and system overview

challenges such as those mentioned above when running the CiteSeer^X digital library. We describe how CiteSeer^X automatically integrates scholarly documents from across the Web and extracts information from them; how we deal with document clustering, entity linking and name disambiguation; and our policies and experiences in sharing data. Furthermore, we discuss some potential research areas of interest for the future. In making these contributions, the rest of this paper is structured as follows. Section II briefly describes the CiteSeer^X architecture and provides some collection and usage statistics followed by Section III, which describes the data collection process. Section IV describes some of the different types of information extraction we perform followed by Section V where we describe how we deal with issues such as de-duplication, document clustering, entity linking and entity disambiguation. Section VI describes our experiences and policies for sharing data and code, and making services available. Lastly, conclusions and opportunities for future research are discussed in Section VII.

II. CiteSeer^X

CiteSeer^X is a digital library and search engine for academic documents. Traditionally, the focus of CiteSeer^X has been on computer science, information science and related disciplines; however, in recent years there has been an expansion to include other academic fields. The key features of CiteSeer^X are that it automatically performs information extraction on all documents added to the collection and automatic citation indexing, which allows for citations and other information among papers to be linked.

A. System Overview

Figure 1 shows a high-level overview of the CiteSeer^X system architecture. The backend to the system is made up of the crawler, extraction modules and backup data stores. As with most online services, high availability is a challenge and thus the frontend to CiteSeer^X contains load balancers that direct incoming traffic to one of a series of Web servers (three at the time of writing). These Web servers serve the user interface and interact with the repository, database and index servers. The physical system is highly modular and has recently been migrated to a virtual infrastructure that allows for the easy deployment of resources as needed.

TABLE I
COLLECTION AND USAGE STATISTICS FOR CITESEER^X

Statistic	Value
#Documents	3.5 million
#Unique documents	2.5 million
#Citations	80 million
#Authors	3-6 million
#docs added monthly	300,000
#docs downloaded monthly	300,000-2.5 million
Individual Users	800,000
Hits per day	2-4 million

B. Data Stores

As can be seen in Figure 1, CiteSeer^X makes use of three main data stores: the index server that is used to enable fast searching; the master repository that stores the physical files; and the database that stores metadata, citations, and other relevant information. It is important to keep these data stores synchronized and linked since they are often used simultaneously in responding to requests for documents. We link the data stores by means of unique identifiers for each document or cluster of documents. For instance, when a user conducts a search, each result contains a unique Document Object Identifier (DOI). These DOIs are then used to retrieve metadata from the database based on the DOI and download requests are fulfilled based on the DOI. Thus, special care is taken to ensure that these DOIs are consistent across data stores and persistent over time.

C. Collection and Usage Statistics

Table I shows various approximate statistics related to the size of the CiteSeer^X collection as well as its usage.

As can be seen from Table I, CiteSeer^X provides a relatively large collection of scholarly documents with the collection growing by about 10,000 new documents daily of which about 80% are unique. Due to the scale and growth, challenges lie in efficiently extracting information from the documents and scaling to support many users.

III. Data Collection

A central challenge for CiteSeer^X is to aggregate academic papers from across the Web in a single location and process them in a uniform way. Our interest is only in academic papers and thus we make use of focused crawling while filtering out non-academic documents.

A. Focused Crawling

CiteSeer^X uses two instances of Heritrix 1.14 crawler to perform focused crawling. The first crawler is for scheduled crawls while the second is for crawling user submitted URLs. Both crawlers are configured to only save PDF files. The seed URLs of the main crawler are selected from a whitelist [13], which is updated every few months. The whitelist is generated based on crawl history and contains high quality URLs that are most likely to provide plenty of academic PDF links. The average crawling rate varies from 50,000 to 100,000 PDFs per day. Of the crawled documents, about 40% of are eventually identified as being academic and ingested into the database.

B. Document Filtering

During crawling there is no way of knowing whether a PDF we have retrieved is an academic paper or not. Thus we perform document filtering after PDF documents are crawled. We first extract text from all these documents and then classify them using a regular expression that looks for the occurrence of the words *"references"* or *"bibliography."* This is a rather simple and error prone classification scheme and thus we are currently looking into supervised classification methods for academic documents. All documents identified as non-academic are filtered out while all academic papers undergo further metadata extraction and ingestion.

IV. Information Extraction

Information extraction forms a crucial part of CiteSeer$^{\mathcal{X}}$ and affects the overall usability and quality of the service due to the fact that the automatically extracted information is used as the metadata, which is used for searching and interacting with the site and data collection. Due to the fact that CiteSeer$^{\mathcal{X}}$ integrates academic papers from across the Web and is a fully automated system, extraction needs to be robust to variations among paper formats and be scalable. CiteSeer$^{\mathcal{X}}$ currently incorporates several different information extraction modules.

A. Header Extraction

Metadata in the form of information about papers is one of the most important types of information extracted by CiteSeer$^{\mathcal{X}}$. Specifically, CiteSeer$^{\mathcal{X}}$ attempts to extract the following information from each document that passes the document filtering: title, authors, abstract, venue, volume and issue (for journals), page numbers, publisher and publisher address. Extraction is performed using SVMHeaderParse [2], which is an SVM-based header extractor. A recent comparison of various header extraction tools [3] showed that more accurate extraction tools than SVMHeaderParse currently exist; however, it is currently not clear how well those other tools scale, which is an important consideration given the rate at which CiteSeer$^{\mathcal{X}}$ grows.

1) Metadata Corrections and Improvement: It is often not possible to extract all header elements from papers. For instance, preprints may be missing information such as venue and page numbers. Furthermore, there are often errors in extraction [3]. We thus attempt to leverage other sources to improve the quality of our metadata.

a) User Corrections: CiteSeer$^{\mathcal{X}}$ allows users to create accounts through a system known as *MyCiteSeerX* and provide metadata corrections and add additional metadata. A recent log analysis showed that CiteSeer$^{\mathcal{X}}$ receives over 13,000 user corrections per month, though we are yet to analyze the quality of these correction.

b) DBLP: DBLP provides manually curated metadata from publication in computer science and related fields. Since a large proportion of CiteSeer$^{\mathcal{X}}$ papers are from these fields as well, recent efforts have attempted to integrate this metadata through record linking based on matching title and authors from CiteSeer$^{\mathcal{X}}$ metadata with DBLP [4]. It was found that about 25% of papers in CiteSeer$^{\mathcal{X}}$ have matching counterparts in DBLP and on a sample of 1000 randomly selected and labelled papers in CiteSeer$^{\mathcal{X}}$, it was found that precision of 0.75 and recall of 0.8 could be achieved when matching CiteSeer$^{\mathcal{X}}$ papers to DBLP based on the Jaccard similarity of 3-grams in titles.

B. Citation Extraction

Citations play an important role in scholarly documents as they form a graph that can be mined to extract information related to evolution of ideas, importance of work, etc. Citations are extracted for every paper ingested by CiteSeer$^{\mathcal{X}}$ using the ParsCit citation string parsing tool [5]. The section of the text containing citations are first identified from the text based on regular expressions and then each citation is extracted, parsed and tagged. Furthermore, the citation context for each extracted citation is stored, which allows for further citation analysis.

C. Other Information Extraction

Header and citation extraction are the core information extraction modules in CiteSeer$^{\mathcal{X}}$ as they form the majority of metadata that users interact with; however, other information extraction modules also exist that are the output ongoing research in scholarly information extraction.

1) Table Extraction: Tables are common in scholarly documents as a method of summarizing information and findings. CiteSeer$^{\mathcal{X}}$ includes a table extraction module that automatically extracts tables and their associated metadata from documents and allows for tables to be searched for [6]. CiteSeer$^{\mathcal{X}}$ also includes a custom table ranking function *TableRank* that ranks tables by <*query,table*> pairs rather than <*query,document*> pairs, thus preventing many false positives that would arise in regular Web search. This table search functionality with *TableRank* is integrated into the main CiteSeer$^{\mathcal{X}}$ interface.

2) Figure Extraction: Figures are used in academic documents to report experimental results, system architecture and various other things. In [16], authors pointed that this rich information resource have been neglected for a long time in digital libraries. Our current work in this domain consists of:

- Extraction of figure and associated metadata (caption and mention) from digital documents [7]: This work showed that positional and font related information extracted from digital documents can be used for accurate extraction of figure metadata. As extraction of such information is heavily dependent on underlying PDF processing library (such as PDFBox), later a machine learning based system was also developed which uses just textual features.
- A search engine focused on figure related textual metadata extracted from documents [8]: Figure metadata extracted from 160,000 chemical journals were indexed using Lucene. The standard ranking function of Lucene was modified to improve the ranking procedure. This search engine can be readily integrated into the other such functionalities such as table search or author search.

- A complete pipeline for extraction of data from 2D line graphs and scatter plots [15] that consists of 1) classification of figures into 2D plots and others 2) segmentation of 2D plots into X-axis, Y-axis and plot regions, 3) identification of the X-axis, Y-axis labels and legends in the plot and 4) identification of data points (possibly overlapping) for scatter plots and curves for 2D line graphs.

Future work in this area seeks to explore extraction of vector images, understanding the semantics of figures in scholarly documents and extending the data extraction process to other type of graphs.

3) Algorithm Extraction: As mentioned, CiteSeer$^{\mathcal{X}}$ has its roots in computer science and related fields. These fields often contain descriptions of algorithms and thus a module for algorithm extraction was developed [9]. Similar to table and figure extraction, this is an extraction module with specific application to scholarly documents; however, in this case the application is even more specific since it only applies in fields that describe algorithms in papers. As with table extraction, the algorithm search feature is integrated into the main CiteSeer$^{\mathcal{X}}$ interface.

4) Acknowledgment Extraction: Acknowledgments are argued to be an important attribution of gratitude that in some cases refers to at least as much contribution to the paper as that of the last author [10]. In certain domains, such as medicine, where clinical studies span multiple centers, it is not feasible to fit the number of authors in the authors section. Thus, the names of centers are shown as the authors while the individuals contributing to the study are listed in the acknowledgments section. Therefore, CiteSeer$^{\mathcal{X}}$ extracts the acknowledgments sections from research papers into a stand-alone system called *AckSeer*. An ensemble of named entity recognizers is used to extract the entities from the acknowledgments section of a paper and three types of entities are extracted: person names, organizations, and companies. This introduces a problem of identifying different variations of names for the same entity. Formally, it is an entity resolution problem where multiple entities can be clustered under the same canonical name. To do this, AckSeer utilizes a novel search engine based algorithm for disambiguating the plethora of entities found in the acknowledgment sections of papers [11].

D. Challenges for Scholarly Information Extraction

Challenges in scholarly information extraction include accuracy, coverage and scalability. The first two of these challenges apply to scholarly information extraction in general, while the last is mainly specific to big data scholarly information extraction. By accuracy, we mean that when information is extracted, it is extracted correctly. As already mentioned, this can sometimes be challenging due to missing information as well as errors in the extraction itself. As discussed in this Section IV-A1, CiteSeer$^{\mathcal{X}}$ attempts to address this by integrating data from additional sources and using it to improve metadata. In Section V-B, we will describe how the way in which we link and cluster citations can also be used to improve metadata. Coverage is related to the classic tradeoff between precision and recall in information retrieval. When extracting entities such as tables, figures and algorithms it is desirable to achieve good recall by extracting as many of them as possible while also maintaining good precision by not extracting false positives. Lastly, from the perspective of scholarly big data, it is desirable to meet the above goals while processing large amounts of data in a reasonable time. One way of doing this is to make use of efficient algorithms, possibly at the cost of accuracy, precision and recall. Another way is to make use of parallel and distributed processing, such as the map-reduce framework [12].

V. Document Clustering and Entity Linking

Once information has been extracted from documents, the second challenge lies in linking the new data to existing data. In CiteSeer$^{\mathcal{X}}$ this takes the form of de-duplication and clustering, citation linking and matching, and author disambiguation.

A. De-duplication and Clustering

It is reasonable to expect that multiple versions of papers exist on the Web. For instance, the co-authors of a paper might each put a copy of a paper on their personal Website, or a pre-print might exist on a author's website while the final version of a paper exists at a publisher. Furthermore, these multiple versions may not be bitwise identical but may have minor differences (such as the omission of copyright notices) thus making them near duplicates. During ingestion, bitwise identical papers are identified by a SHA1 or MD5 hash and discarded; however, near duplicates are retained and clustered. CiteSeer$^{\mathcal{X}}$performs clustering of near duplicates based on the metadata extracted from each document and attempts to match the extracted metadata with the metadata of already ingested papers. When a match is found, the papers are assigned to the same *cluster* where a cluster represents a set of near duplicates of the same paper. When no match is found, a new cluster is created and the new paper is added as the only member of that cluster. One of the shortcomings of the metadata based clustering is that the performance is directly related to the quality of the automatically extracted metadata. Thus, we are currently investigating the use of near duplicate detection methods based on the full text of articles [14].

B. Citation Clustering and Linking

Many scholarly documents might cite the same influential papers and therefore CiteSeer$^{\mathcal{X}}$ performs citation clustering whereby citations to the same paper are grouped together in the same *cluster*. The method for doing this is the same as the document clustering method and is based on the metadata extracted from each citation during the citation string parsing. The citation clusters are in fact the same clusters as used for documents, with a flag to indicate whether or not CiteSeer$^{\mathcal{X}}$ contains a version of the paper or just a citation to it. Furthermore, for each citation C extracted from a paper P a <*cluster(P), cites, cluster(C)*> relation is created thereby

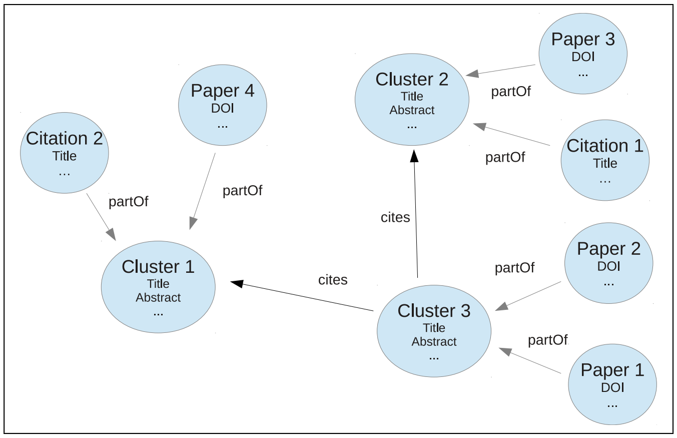

Fig. 2. Cluster, document and citation graph

creating a citation graph. Figure 2 shows an example of the relationship between clusters, documents and citations.

As can be seen from the figure, clusters contain both papers and citations, which have been matched to the cluster based on the metadata automatically extracted during header extraction for papers and citation parsing for citations. In the figure, the papers in Cluster 3 cite papers in Clusters 1 and 2 as is demonstrated by the fact that there is a *cites* relationship and the fact that Clusters 1 and 2 both have citation nodes. One of the benefits of linking papers and citations to clusters is that it can be used to improve metadata. For instance, it has already been discussed how certain metadata fields may be missing when performing header extraction from a paper; however, when there is a citation to that paper as well then it is possible to improve the paper metadata by incorporating the fields extracted from the citation.

C. Author Name Disambiguation

Due to the fact that publishers have different formats, author names often have different representations. For instance, *C. L. Giles* and *C. Lee Giles* may be two ways of referring to the same author. The author name disambiguation problem involves identifying whether or not two references are to the same person. The method for author disambiguation currently in use is based on building a similarity profile for each author and disambiguating authors using random forests [17]. CiteSeer$^{\chi}$ maintains a page for each disambiguated author that contains different variations of their names, their affiliation and homepage, their H-index based on CiteSeer$^{\chi}$ data and a list of their publications in CiteSeer$^{\chi}$.

VI. Data Sharing

There are many reason to share the data collected by CiteSeer$^{\chi}$, such as to foster collaboration and research. However, challenges exist for sharing our data due to the size of the data as well as issues related to potential copyright violation. Even though CiteSeer$^{\chi}$ data is crawled from the public Web while obeying site crawling policies, it is possible that some copyrighted material is retrieved and from time to time CiteSeer$^{\chi}$ receives requests from authors and publishers to remove documents. However, even with these challenges, we still believe it is beneficial to share data.

One way CiteSeer$^{\chi}$ data is shared in through the Open Archives Initiative Protocol for Metadata Harvesting, which is a standard proposed by The Open Archive Initiative in order to facilitate content dissemination. By programatically accessing the CiteSeer$^{\chi}$ OAI Harvest URL[5], it is possible to download the metadata for all papers that exist in CiteSeer$^{\chi}$. This is the easiest way to access CiteSeer$^{\chi}$ data and seems to be widely used with an average of 4983 requests per month.

From time to time, researchers are interested in more than just the CiteSeer$^{\chi}$ metadata. For instance, during November 2013 CiteSeer$^{\chi}$ received 9 requests for data via the contact form on the CiteSeer$^{\chi}$ Website. In cases like these, we make dumps of our databases available on Amazon S3 for download. This has the benefit for alleviating us of the cost of distributing the data since the cost of download traffic is paid for by the user and not by us.

A challenge still remains however in how to best go about distributing the core repository, which contains the actual PDF papers and extracted text. Besides the copyright issues already spoken about, there are challenges in distributing the data which currently is larger than 6TB. Furthermore, this repository is growing at a rate of about 10–20GB per day thereby making keeping this repository synchronized with others over the Web a challenge.

A. Code Sharing

CiteSeer$^{\chi}$ and related projects (such as the extraction modules) are usually open sourced under the permissive Apache License Version 2. The motivation for doing this is to allow other research groups to run their own versions of CiteSeer$^{\chi}$ as well as to allow the community to make improvements to CiteSeer$^{\chi}$ that can be used to improve the service. The source code for CiteSeer$^{\chi}$ was previously hosted on SourceForge[6]; however, recently the source code has been migrated to GitHub[7] to enable better collaboration. Since making this migration in July 2013, CiteSeer$^{\chi}$ has been forked 5 times.

B. CiteSeerExtractor

CiteSeer$^{\chi}$ runs other services in addition to the OAI service already mentioned to simplify the ways in which people interact with and make use of scholarly big data. CiteSeerExtractor is a stand-alone Web service that provides a RESTful API for information extraction from scholarly documents. Based on the extraction modules currently used in CiteSeer$^{\chi}$, CiteSeerExtractor can be integrated into any application that needs to perform scholarly information extraction. This greatly simplifies the information extraction process and allows for centralized extraction that can easily be improved without needing to distribute the improvements. Like CiteSeer$^{\chi}$, CiteSeerExtractor is an open source project publicly available on our GitHub page, thereby making it easy for other research

[5] http://citeseerx.ist.psu.edu/oai2
[6] http://citeseerx.sourceforge.net/
[7] https://github.com/SeerLabs

groups to deploy their own versions and allows us to benefit from any community improvements to the software.

VII. Conclusions

In this paper, we have provided a case study of CiteSeer$^{\mathcal{X}}$ and how it integrates data from across the Web and performs automatic extraction, clustering, entity linking and name disambiguation on that data. Furthermore, we have described how we share data, code and services and some of the challenges and opportunities that have arisen from doing that. All of these tasks are interesting in and of themselves from a research perspective; however, recently we have been faced with the pressure of not only supporting these sorts of operations, but also the need to do so in scalable ways. In this regard, as CiteSeer$^{\mathcal{X}}$ has continued to grow, we have been faced with increasing pressure to be able to deal with the needs and demands of big data and the need to design algorithms, systems, and processes that are scalable. Though not discussed in this paper, the first step in doing this has involved migrating CiteSeer$^{\mathcal{X}}$ from a physical architecture to a virtual architecture. This has the main benefit of allowing us to provision resources on demand in order to meet increasing requirements for processing power. However, in addition to this we see a number of other research opportunities in dealing with scholarly big data.

It was mentioned in Section III that we crawl between 50,000 and 100,000 PDFs per day of which about 40% are scholarly documents; however, we only ingest about 10,000 new documents per day and thus a bottleneck exists in our ingestion pipeline. Research opportunities thus potentially exists in distributed ingestion.

Opportunities also exist in investigating new information extraction methods that can be applied to scholarly data, such as extracting data from figures as previously mentioned. This is one of the main benefits of automated scholarly big data management systems since new extraction modules can easily be added. As already mentioned though, there are often errors in extraction, thus there is the potential to identify new sources of information that can be integrated into CiteSeer$^{\mathcal{X}}$that can be used to improve the quality of automatically extracted data. Furthermore, the user corrections submitted through *MyCiteSeerX* provides a useful set of data that can be used to try and better understand where automatic extraction fails, which may be beneficial for improving the extraction algorithms.

Lastly, given the large number of hits that CiteSeer$^{\mathcal{X}}$ receives per day, we believe that there is an opportunity in analyzing access logs in order to determine how best to provision services and design algorithms. For instance, logs analysis can be used to gain insight into user requests and requirements, improve ranking and determine which data should be cached.

Acknowledgments

This work is partially supported by the National Science Foundation.

References

[1] E. Archambault, D. Amyot, P. Deschamps, A. Nicol, L. Rebout, and G. Roberge, "Proportion of Open Access Peer-Reviewed Papers at the European and World Levels - 2004-2011," European Commission DG Research & Innovation. August, 2013.

[2] H. Han, C. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. Fox, "Automatic document metadata extraction using support vector machines," in *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, 2003, pp. 37–48.

[3] M. Lipinski, K. Yao, C. Breitinger, J. Beel, and B. Gipp, "Evaluation of header metadata extraction approaches and tools for scientific PDF documents," *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries - JCDL '13*, pp. 385–486, 2013.

[4] C. Caragea, J. Wu, A. Ciobanu, K. Williams, H.-H. Fernandez-Ramirez, Juan Chen, Z. Wu, and C. L. Giles, "CiteSeerX: A Scholarly Big Dataset," in *36th European Conference on Information Retrieval (To Appear)*, 2013.

[5] I. Councill, C. Giles, and M. Kan, "ParsCit: an Open-source CRF Reference String Parsing Package." *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, 2008.

[6] Y. Liu, K. Bai, P. Mitra, and C. Giles, "Tableseer: automatic table metadata extraction and searching in digital libraries," *Proceeding of the 7th annual international ACM/IEEE joint conference on Digital libraries - JCDL '07*, 2007, pp. 91–10.

[7] S. R. Choudhury, P. Mitra, A. Kirk, S. Szep, D. Pellegrino, S. Jones, and C. L. Giles, "Figure metadata extraction from digital documents," in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 135–139.

[8] S. R. Choudhury, S. Tuarob, P. Mitra, L. Rokach, A. Kirk, S. Szep, D. Pellegrino, S. Jones, and C. L. Giles, "A figure search engine architecture for a chemistry digital library," in *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2013, pp. 369–370.

[9] S. Carman, "AlgSeer: An Architecture for Extraction, Indexing and Search of Algorithms in Scientific Literature, *MSc Thesis, The Pennsylvania State University*, 2013.

[10] M. Khabsa, P. Treeratpituk, and C. L. Giles, "Ackseer: a repository and search engine for automatically extracted acknowledgments from digital libraries," in *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*. ACM, 2012, pp. 185–194.

[11] M. Khabsa, P. Treeratpituk, and C. L. Giles, "Entity resolution using search engine results," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 2363–2366.

[12] J. Dean and S. Ghemawat, "MapReduce," *Communications of the ACM*, vol. 51, no. 1, p. 107, Jan. 2008.

[13] J. Wu, P. Teregowda, J. Ramírezm P. Mitra, S. Zheng and C. L. Giles, "The evolution of a crawling strategy for an academic document search engine: whitelists and blacklists," in *Proceedings of the 3rd Annual ACM Web Science Conference - WebSci '12*, 2012, pp. 340–343.

[14] K. Williams and C. L. Giles, "Near duplicate detection in an academic digital library," in *Proceedings of the 2013 ACM symposium on Document engineering - DocEng '13*. 2013, pp. 91–94.

[15] X. Lu, S. Kataria, W. J. Brouwer, J. Wang, P. Mitra and C. L. Giles, "Automated analysis of images in documents for intelligent document search", *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 12, no. 2, p. 65–81, 2009.

[16] S. Carberry, S. Elzer and S. Demir, "Information graphics: an untapped resource for digital libraries", in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 581–588.

[17] P. Treeratpituk and C. L. Giles, "Disambiguating authors in academic publications using random forests, in *Proceedings of the 2009 joint international conference on Digital libraries - JCDL '09*, 2009, pp. 39–48.

[18] D. Laney, "3D Data Management: Controlling Data Volume, Velocity, and Variety", *Application Delivery Strategies*, META Group Inc, 2001.

[19] P. Larsen amd M. von Ins, "The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index" *Scientometrics*, vol. 84, no. 3, p. 575–603, 2010.