Big Scholarly Data in CiteSeerX: Information Extraction from the Web

Alexander G. Ororbia II
Pennsylvania State University
University Park, PA, 16802
ago109@ist.psu.edu

Jian Wu
Pennsylvania State University
University Park, PA, 16802
jxw394@ist.psu.edu

Madian Khabsa
Pennsylvania State University
University Park, PA, 16802
madian@psu.edu

Kyle Williams
Pennsylvania State University
University Park, PA, 16802
kwilliams@psu.edu

C. Lee Giles
Pennsylvania State University
University Park, PA, 16802
giles@ist.psu.edu

ABSTRACT

We examine CiteSeerX, an intelligent system designed with the goal of automatically acquiring and organizing large-scale collections of scholarly documents from the world wide web. From the perspective of automatic information extraction and modes of alternative search, we examine various functional aspects of this complex system in order to investigate and explore ongoing and future research developments[1].

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: General; H.3 [**Information Storage and Retrieval**]: Digital Libraries

General Terms

Information Extraction, Web Crawling, Intelligent Systems, Digital Libraries, Scholarly Big Data

Keywords

scholarly big data; citeseerx; information acquisition and extraction; digital library search engine; intelligent systems

1. INTRODUCTION

Large-scale scholarly data is the, "...vast quantity of data that is related to scholarly undertaking" [27], much of which is available on the World Wide Web. It is estimated that

---

[1]An earlier version of this paper was accepted at the 4th Workshop on Automated Knowledge Base Construction (AKBC) 2014 workshop.

there are at least 114 million English scholarly documents or their records[2] accessible on the Web [15].

In order to provide convenient access to this web-based data, intelligent systems, such as CiteSeerX, are developed to construct a knowledge base from this unstructured information. CiteSeerX does this autonomously, even leveraging utility-based feedback control to minimize computational resource usage and incorporate user input to correct automatically extracted metadata [28]. The rich metadata that CiteSeerX extracts has been used for many data mining projects. CiteSeerX provides free access to over 4 million full-text academic documents and rarely seen functionalities, e.g., table search.

In this brief paper, after a brief architectural overview of the CiteSeerXsystem, we highlight several CiteSeerX-driven research developments that have enabled such a complex system to aid researchers' search for academic information. Furthermore, we look to the future and discuss investigations underway to further improve CiteSeerX's ability to extract information from the Web and generate new knowledge.

2. OVERVIEW: ARCHITECTURE

While major engines, such as Microsoft Academic Search and Google Scholar, and online digital repositories, such as DBLP, provide publication and bibliographic collections or metadata of their own, CiteSeerX stands in contrast for a variety of reasons. CiteSeerX has proven to be a rich source of scholarly information beyond publications as exemplified through various derived data-sets, ranging from citation graphs to publication acknowledgments [16], meant to aid academic content management and analysis research [1]. Furthermore, CiteSeerX's open-source nature allows easy access to its implementations of tools that span focused web crawling to record linkage [35] to meta-data extraction to leveraging user-provided meta-data corrections [31]. A key aspect of CiteSeerX 's future lies in not only serving as an engine for continuously building an ever-improving collection of scholarly knowledge at web-scale, but also as a set of publicly-available tools to aid those interested in building digital library and search engine systems of their own.

---

[2]Scholarly documents are defined as journal & conference papers, dissertations & masters theses, academic books, technical reports, and working papers. Patents are excluded.
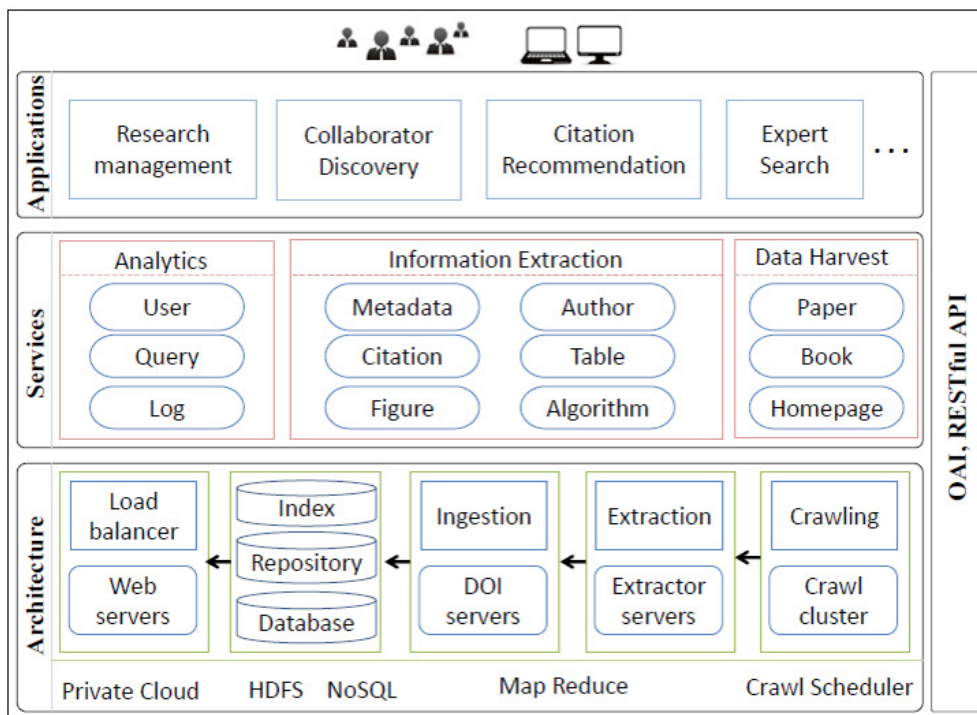
Applications

| Research management | Collaborator Discovery | Citation Recommendation | Expert Search | ... |

Services

Analytics: User, Query, Log

Information Extraction: Metadata, Author, Citation, Table, Figure, Algorithm

Data Harvest: Paper, Book, Homepage

Architecture

Load balancer, Web servers ← Index, Repository, Database ← Ingestion, DOI servers ← Extraction, Extractor servers ← Crawling, Crawl cluster

Private Cloud   HDFS   NoSQL   Map Reduce   Crawl Scheduler

OAI, RESTful API

Figure 1: High-level overview of the CiteSeerX platform.

CiteSeerX can be compactly described as a 3-layer complex system (as shown in Figure 1[3]). The architecture layer demonstrates the high-level system modules as well as the work flow. It can be divided into two parts: the frontend (Web servers and load balancers) that interacts with users, processes queries, and provides different web services; the backend (crawler, extraction, and ingestion) that performs data acquisition, information extraction and supplies new data to the frontend. The services layer provides various services for either internal or external applications by APIs. The applications layer lists scholarly applications that build upon these services.

The coupled efforts of the various modules that compose the system pipeline facilitate the complex processing required for extracting and organizing the unstructured data of the web. While the architecture consists of many modules[4], in subsequent sections we will sample technologies representative of CiteSeerX 's knowledge-generation process as well as expand on future directions for these modules to improve CiteSeerX 's ability to harvest knowledge from the web throughout.

3. DATA ACQUISITION

Automatic web crawling agents compose CiteSeerX's front-line for information gathering. Some agents are used in scheduled re-crawls using a pre-selected whitelist of URL's [30] to improve the freshness of CiteSeerX's data while oth-

ers are pointed to begin crawling newly discovered locations (some provided by users). Only PDF documents are imported into the crawl repository and database using the CiteSeerX crawl document importer [29], which are further examined by a filter. Currently, a rule-based filter is used to determine if documents crawled by these agents are academic or not[5]. To improve performance and escape limitations of this current system, we are developing a more sophisticated document filter that utilizes structural features to contruct a discriminative model for classifying documents [8]. We have considered various types of structural features, ranging from file specific, e.g., file size, page count, to text specific, e.g., line-length.

This structure-based classification algorithm was evaluated on large sets of manually labelled samples, randomly drawn from the web crawl repository and the CiteSeerX production repository. Results indicate that the Support Vector Machine (SVM) [9] achieves the highest precision (88.9%), F-measure (85.4%) and accuracy (88.11%), followed by the Logistic Regression classifier [2], achieving a slightly lower precision (88.0%), F-measure (81.3%) and accuracy (87.39%). These models, in tandem with feature-engineering, significantly outperform the rule-based baseline. Further work will extend this binary classifier to classify documents into multiple categories such as papers, reports, books, slides, and posters, which can facilitate category-dependent metadata extraction, and document alignments, e.g., papers and slides.

While our discriminative model (i.e., a supervised model trained to perform academic document classification) is designed to better filter document data harvested by Cite-

---

[3]Note: This overview displays the proposed, private cloud-based CiteSeerX platform described in [35], though several modules described in this paper exist in the current Cite-SeerX system or are under active development.

[4]For more detailed explications of the CiteSeerX architecture itself, we refer the reader to [35, 31].

---

[5]80–90% precision and 70–80% recall on a manually labelled document set

SeerX's web crawler – *citeseerxbot*, a similar approach could also be used to construct "exploratory", topical crawling agents. Such agents could sift through the web information, discerning relevant resources, perhaps intelligently navigating target websites, and making decisions in partially observable environments. Professional researcher homepages could be fruitful sources of academic publications [12]. Such a functionality can be achieved by automatic classification of web pages based on both the URL and crawl history.

4. INFORMATION EXTRACTION

Once relevant information has been harvested, the information must be processed and organized to facilitate search-based services and scholarly applications. Textual content is extracted from these files, including document metadata such as the header, the body, and references. This metadata, which is valuable knowledge in of itself, is then automatically indexed for searching, clustering, and other purposes.

4.1 Document De-duplication

In submission-based digital libraries, such as arXiv and ACM Digital Library, duplication is rare. For a crawl-based digital library, such as CiteSeerX and Google Scholar, document duplication is inevitable but can be handled intelligently. Two types of duplication are considered: bitwise- and near-duplication. Bitwise duplicates occur in web crawling and ingestion, detected by matching SHA1 values of new documents against the existing ones in the database. Once detected, these documents are immediately removed. Near-duplication (ND) refers to documents with similar content but minor differences.

Intuitively, ND documents can be detected if two papers have the same title and author names. However, matching title/author strings directly is inefficient because it is very sensitive to the parsing results, which are often noisy. In CiteSeerX, the ND is detected using a key mapping algorithm. A set of keys are generated for a new document when it is ingested. These document keys are built by concatenating normalized author last names and normalized titles.

Document clusters are modified when a user correction occurs. When a user corrects paper metadata, it is removed from its existing cluster and assigned to a new cluster based on the new metadata. Should the cluster it previously belonged to become empty, then that cluster is deleted. The citation graph is generated when papers are ingested. In this graph, the nodes are document clusters and each directional edge represents a citation relationship. This notion of document cluster integrates papers and citations, making it more convenient to perform statistical calculations, ranking, and network analysis.

4.2 Header Extraction

Headers, which contain useful information fields such as paper title and author names, are extracted using SVM-HeaderParse [13], which is a SVM-based header extractor. This model first extracts features from textual content extracted from a PDF document, which is done using a rule-based, context-dependent word clustering method for word-specific feature generation, with the rules extracted from various domain databases and text orthographic properties of words, e.g., capitalization. Following this, independent line classification is performed, where a set of "one-vs-others" classifiers are trained to associate lines to specific target vari-

ables. Lastly, a contextual line classification step is executed, which entails encoding the context of these lines, i.e., N lines before and after a target line labelled from the previous step, as binary features to construct context-augmented feature representations. Header metadata, such as author names, is then extracted from these classified lines. Evaluation is performed using 500 labelled examples of headers of computer science papers. On 435 unseen test samples, the model achieves a 92.9% accuracy and ultimately outperforms a Hidden Markov Model in most other performance metrics.

While SVMHeaderParse achieves satisfying results for high-quality text files converted from academic papers in computer science, its performance in other subject domains is relatively low. A recent study by Lipinski et al. compared several header parsers based on a sample of arXiv papers, and found that the GROBID [20] header extractor outperforms all its competitors. Given that the metadata quality can be improved by 20%, this model becomes a good candidate for the replacement of SVMHeaderParse. The improved quality of title and author information is especially important for extracting accurate metadata in other fields through paper-citation alignment as well as for cleaning metadata using high quality reference data (see below).

However, aside from addressing the issue of training a purely supervised model on high-quality extracted headers (which may not reflect a noisier reality), we desire a more general classifier. With respect to this, we are developing a multi-channel discriminative model capable of performing multiple, domain-dependent classification tasks, for example, using different extraction models for computer science and physics papers.

4.3 Extracting Citations

CiteSeerX uses ParsCit [11] for citation extraction, which has a conditional random field model core [17] for labelling token sequences in reference strings. This core was wrapped by a heuristic model with added functionality to identify reference string locations from plain text files. Furthermore, based on a reference marker (marked or unmarked depending on citation style), ParsCit extracts citation context by scanning a body text to find citations that match a specific reference string, which is valuable for users interested in seeing what authors say about a specific article.

Evaluations were performed on three datasets: Cora (S99), CiteSeerX, and FLUX-CiM [10]. The results show that the core module of ParsCit that performs reference string segmentation performs satisfactorily and is comparable to the original CRF based system in [22]. ParsCit outperforms FLUX-CiM, which is an unsupervised reference string parsing system, on the key common fields of "author" and "title". However, ParsCit makes some mistakes such as not segmenting "volume", "number" and "pages" given that it currently does not further tokenize beyond white spaces (e.g., "11(4):11-22" versus "11 (4) 11 - 22"). We intend to incorporate some preprocessing heuristics to ParsCit to correct for these errors.

4.4 Aligning Papers and Citations

Among all header fields, the title, authors, and abstract are usually present on the front page (not necessarily the first page) of most academic papers. They are also relatively

easy to extract due to similar layouts across different paper templates.

In contrast, the date, venue, and publication information do not always appear on the front page. Even if they do, it is non-trivial to extract them due to significant variance among paper templates. Nonetheless, these fields are usually arranged in a structured format in the citation string. Therefore, we align papers and citations and adopt values of these fields from citation parsing results.

The alignment between papers and citations is implemented via a key-mapping algorithm, in which a paper and a citation match if they have the same keys, constructed by concatenating normalized title and author strings. Aligning papers and citations is helpful for retrieving accurate venue and date information, which is further used to calculate the venue impact factor.

4.5 Disambiguating Authors

In addition to document search, CiteSeerX allows users to search for an author's basic information and previous publications where a typical query string is an author name. However, processing a name-based query is complex given that different authors may share the same name. In a collection containing many millions of papers and un-disambiguated authors, using a distance function to compare author similarity would require $\mathcal{O}(n^2)$ time complexity and thus intractable for large n.

To reduce the number of comparisons, CiteSeerX groups names into small blocks and claims that an author can only have different name variations within the same block. This reduces the problem to checking pairs of names within the same block. CiteSeerX groups two names into one block if the last names are the same and the first initials are the same. Leveraging extra author information, CiteSeerX uses a hybrid DBSCAN and Random Forest model to resolve any ambiguities [23].

4.6 Cleaning Metadata

Metadata cleaning involves detecting incorrectly extracted metadata, and then correcting them. One common approach is to match the target metadata against a reference database using one or multiple keys, and replace all or suspicous metadata with their counterparts in the reference database. For a system such as CiteSeerX, the metadata are extracted from documents coming from various sources which are noisy. It is feasible to improve metadata quality using submission-based digital libraries, e.g., DBLP, given that a large proportion of CiteSeerX papers are from the same subject domains.

Recently, Caragea et al. attempted to integrate CiteSeerX citation context into DBLP metadata by matching titles and authors of these two data sets. They found that $\sim 25\%$ of CiteSeerX papers have matching counterparts in DBLP with 80% recall and 75% precision[6]. Higher precision may be achieved at the cost of a relatively low recall, but this provides a promising way of acquring reliable metadata for a considerable proportion of CiteSeerX papers. By adopting metadata from other digital libraries, i.e., PubMed or IEEE, more incorrectedly extracted metadata can be corrected.

It is also feasible to clean paper titles by leveraging commercial search engines, such as Google and Bing. These giant search engines, by applying their own proprietary document parsers, are usually able to retrieve metadata more accurately, especially paper titles. This can be achieved by submitting API requests containing CiteSeerX paper ID's and parsing the response pages. However, these APIs usually only have limited access, so it is desirable to prioritize papers with ill-conditioned metadata.

A fraction of such papers can be found out by comparing the downloading rate and citation rate. Log analysis showed a positive correlation between these two numbers for papers with normal metadata. Given this correlation, the fact that a certain highly downloaded paper has zero citation rate is an indicator of ill-conditioned metadata. Manual inspection of these papers found that many had their header metadata incorrectly extracted, which resulted in mis-assigned citations. These papers can then be assessed using commercial search engine APIs for possible corrections.

In addition, CiteSeerX leverages additional author information (i.e., affiliation, email, paper key-phrases and topics) to improve its disambiguation process. To resolve any inconsistent cases (results that violate the transitivity principle), DBSCAN, a clustering algorithm based on the density reachability of data points, is applied in conjunction with a Random Forest ensemble efficiently trained to handle ambiguous cases found at cluster boundaries [23].

5. FACILITATING ALTERNATIVE FORMS OF SEARCH

CiteSeerX aims to provide important alternative means of exploring scholarly data, beyond traditional author or title-based query. In order to facilitate such depth, other informative aspects of publications, namely algorithmic psuedocode and scientific figures, must be treated as potential target metadata. While these pose greater challenge for content processing, extracting and indexing unique document components may yield intriguing ways of gathering related documents based on non-conventional criterion.

5.1 Algorithm Extraction and Mining

Algorithms are ubiquitous in computer science and the related literature that offer stepwise instructions for solving computational problems. With new algorithms being reported every year, it would be useful for CiteSeerX to offer services that automatically identify, extract, index, and search an ever-increasing collection of algorithms, both new and old. Such services could serve researchers and software developers looking for cutting-edge solutions to their daily technical problems.

A majority of algorithms in computer science documents are summarized/represented as pseudo-code [24]. Three methods for detecting pseudo-code in scholarly documents include rule based, machine learning based, and combined methods. We found that combined methods perform the best (with respect to F1 score) for extracting indexable metadata (captions, textual summaries) for each detected pseudo-code. On the other hand, extracting algorithm-specific metadata (such as algorithm name, target problems, or run-time complexity) proves to be more challenging given differing algorithm-writing styles and the presence of multiple algorithms in one paper (requiring disambiguation).

Knowing what an algorithm actually does could shed light on multiple applications such as algorithm recommendation

---

[6]The matching rate is limited by metadata quality. In addition, CiteSeerX indexes a large fraction of non-computer science papers.

and ranking. We have begun exploring the mining of algorithm semantics by studying the algorithm co-citation network [25] and are continuing to study how algorithms influence each other over time. A temporal study would allow us to discover new and influential algorithms and also learn how existing algorithms are applied in various fields of study. In order to do this, we propose the construction and analysis of the algorithm citation network, where each node is an algorithm, and each direct edge represents how an algorithm uses another existing algorithm. With respect to this goal, we are building a discriminative model to classify algorithm citation contexts [26], which would then allow for automatic construction of a large scale algorithm citation network.

5.2 Figure Extraction

Academic papers usually contain figures that report experimental results or system architecture(s). Often, the data present in such figures cannot be found within the text. Thus, extracting figures and associated information would be useful in better understanding content. However, it is non-trivial to extract vector graphics (SVG, eps) from PDF documents as these contain drawing instructions that are interleaved in the PDF document and thus difficult to discern from other non-figure drawing instructions.

[4] proposed an approach for figure extraction from PDF documents, where each page is converted into an image and then analyzed through segmentation algorithms to detect text and graphics regions. This approach was improved using clustering, heuristics for extracting positional and font-related features, and a machine learning-based system that leveraged syntactic features [7].

In addition to figure metadata, we have also attempted to extract information from the figures themselves, a problem for which only limited success has been previously reported [21]. We focused on analyzing line graphs, given their highly frequent usage in research papers to report results. Following the work of [21], we were able to develop a classification algorithm for classifying a figure as a line graph or not, and obtained 85% accuracy using a set of 475 figures. Future work will involve extraction of curves from plotting regions.

6. CONCLUSION

In this paper, we described CiteSeerX and discussed the various aspects of this complex system that comprise its data gathering and information extraction process. In particular, we examined the system from the perspective of comparing current implementations with future directions . Through a pipeline of automatic mechanisms, CiteSeerX harvests scholarly data from the world wide web and parses and cleans this information to extract critical content, such as publication metadata and citation information, useful for document curation and knowledge organization. Much of this information is difficult to extract and requires the use of computational intelligence to filter and process documents in a variety of ways, mining even items such as algorithms and figures, to facilitate novel investigation of the data. As we have shown in our research, these aspects of scholarly data and the CiteSeerX-generated metadata facilitate analysis at the macro- and micro-levels.

Taking advantage of the rich information foundation created by CiteSeerX, we have built a variety of scholarly applications to generate additional knowledge that can be used to analyze and explore scholarly documents and the na-

ture of academia. These include *RefSeer*[7] for recommending topic and context-related citations given a portion of a paper [14], *CollabSeer*[8] for discovering potential collaborators for a given author [5], and *CSSeer*[9], a Computer Science expert discovery and related topic recommendation system [6].

Other aspects of scholarly data that CiteSeerX handles include tables [19], acknowledgements [16], table of contents [34], and back-of-the-book indices [33, 32]. It could prove to be an interesting and useful task to build query functionality for these information units to allow for yet even deeper exploration of large-scale scholarly data. Through future experimental and innovation, the CiteSeerX system can be used to effectively decompose scholarly data to its fundamental details, all of which forward the scientific endeavor of large-scale knowledge discovery and creation.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] S. Bhatia, C. Caragea, H.-H. Chen, J. Wu, P. Treeratpituk, Z. Wu, M. Khabsa, P. Mitra, and C. L. Giles. Specialized research datasets in the CiteSeerX digital library. In *D-Lib Magazine*, volume 18, 2012.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[3] C. Caragea, J. Wu, A. Ciobanu, K. Williams, J. Fernandez-Ramirez, H.-H. Chen, Z. Wu, and C. L. Giles. Citeseerx: A scholarly big dataset. ECIR '14, pages 311–322, 2014.

[4] H. Chao and J. Fan. Layout and content extraction for pdf documents. In *Document Analysis Systems VI*, pages 213–224. Springer, 2004.

[5] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles. Collabseer: a search engine for collaboration discovery. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 231–240. ACM, 2011.

[6] H.-H. Chen, P. Treeratpituk, P. Mitra, and C. L. Giles. CSSeer: An expert recommendation system based on CiteseerX. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '13, pages 381–382. ACM, 2013.

[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 *Proceedings of ICDAR*, pages 135–139. IEEE, 2013.

[8] Cornelia Caragea, Jian Wu and C. L. Giles. Automatic identification of research articles from crawled documents. In *Proceedings of WSDM-WSCBD*, 2014.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[10] E. Cortez, A. S. da Silva, M. A. Gonçalves, F. Mesquita, and E. S. de Moura. Flux-cim: Flexible

---

[7]http://refseer.ist.psu.edu/

[8]http://collabseer.ist.psu.edu/

[9]http://csseer.ist.psu.edu/

unsupervised extraction of citation metadata. JCDL '07, pages 215–224, 2007.

[11] I. G. Councill, C. L. Giles, and M.-Y. Kan. Parscit: an open-source crf reference string parsing package. LREC '08, 2008.

[12] S. D. Gollapalli, C. L. Giles, P. Mitra, and C. Caragea. On identifying academic homepages for digital libraries. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, JCDL '11, pages 123–132, New York, NY, USA, 2011. ACM.

[13] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines. In *Digital Libraries, 2003. Proceedings. 2003 Joint Conference on*, pages 37–48. IEEE, 2003.

[14] W. Huang, S. Kataria, C. Caragea, P. Mitra, C. L. Giles, and L. Rokach. Recommending citations: translating papers into references. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1910–1914. ACM, 2012.

[15] M. Khabsa and C. L. Giles. The number of scholarly documents on the public web. *PloS one*, 9(5):e93949, 2014.

[16] 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*, pages 185–194. ACM, 2012.

[17] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ICML '01, pages 282–289, 2001.

[18] M. Lipinski, K. Yao, C. Breitinger, J. Beel, and B. Gipp. Evaluation of header metadata extraction approaches and tools for scientific pdf documents. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '13, pages 385–386, New York, NY, USA, 2013. ACM.

[19] Y. Liu, K. Bai, P. Mitra, and C. L. Giles. TableSeer: Automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '07, pages 91–100. ACM, 2007.

[20] P. Lopez. GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL'09, pages 473–474, Berlin, Heidelberg, 2009. Springer-Verlag.

[21] X. Lu, S. Kataria, W. J. Brouwer, J. Z. Wang, P. Mitra, and C. L. Giles. Automated analysis of images in documents for intelligent document search. *IJDAR*, 12(2):65–81, 2009.

[22] F. Peng and A. McCallum. Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, 42(4):963–979, July 2006.

[23] P. Treeratpituk and C. L. Giles. Disambiguating authors in academic publications using random forests. JCDL '09, pages 39–48, 2009.

[24] S. Tuarob, S. Bhatia, P. Mitra, and C. Giles. Automatic detection of pseudocodes in scholarly documents using machine learning. In *Proceedings of ICDAR*, 2013.

[25] S. Tuarob, P. Mitra, and C. L. Giles. Improving algorithm search using the algorithm co-citation network. In *Proceedings of JCDL*, pages 277–280, 2012.

[26] S. Tuarob, P. Mitra, and C. L. Giles. A classification scheme for algorithm citation function in scholarly works. In *Proceedings of JCDL*, JCDL '13, pages 367–368, 2013.

[27] K. Williams, J. Wu, S. R. Choudhury, M. Khabsa, and C. L. Giles. Scholarly big data information extraction and integration in the CiteSeerX digital library. In *Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on*, pages 68–73. IEEE, 2014b.

[28] J. Wu, A. Ororbia, K. Williams, M. Khabsa, Z. Wu, and C. L. Giles. Utility-based control feedback in a digital library search engine: Cases in CiteSeerX. In *9th International Workshop on Feedback Computing (Feedback Computing 14)*. USENIX Association, 2014.

[29] J. Wu, P. Teregowda, M. Khabsa, S. Carman, D. Jordan, J. San Pedro Wandelmer, X. Lu, P. Mitra, and C. L. Giles. Web crawler middleware for search engine digital libraries: a case study for citeseerx. WIDM '12, pages 57–64, New York, NY, USA, 2012. ACM.

[30] J. Wu, P. Teregowda, J. P. F. Ramírez, 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, pages 340–343, New York, NY, USA, 2012. ACM.

[31] J. Wu, K. Williams, H.-H. Chen, M. Khabsa, C. Caragea, A. Ororbia, D. Jordan, and C. L. Giles. Citeseerx: Ai in a digital library search engine. In *The Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence*, IAAI '14, 2014.

[32] Z. Wu and C. L. Giles. Measuring term informativeness in context. In *Proceedings of NAACL-HLT 2013*, page 259-269, 2013.

[33] Z. Wu, Z. Li, P. Mitra, and C. L. Giles. Can back-of-the-book indexes be automatically created? In *Proceedings of CIKM*, pages 1745–1750, 2013.

[34] Z. Wu, P. Mitra, and C. Giles. Table of contents recognition and extraction for heterogeneous book documents. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1205–1209, 2013.

[35] Z. Wu, J. Wu, M. Khabsa, K. Williams, H.-H. Chen, W. Huang, S. Tuarob, S. R. Choudhury, A. Ororbia, P. Mitra, and others. Towards building a scholarly big data platform: Challenges, lessons and opportunities. In *Proceedings of the International Conference on Digital Libraries 2014*, volume 447, page 12. JCDL 2014.