CiteSeerX: Intelligent Information Extraction and Knowledge Creation from Web-Based Data

Alexander G. Ororbia II, Jian Wu, and C. Lee Giles
IST, Pennsylvania State University, University Park, PA, 16802
Email: ago109@ist.psu.edu

1 Introduction

Large-scale scholarly data is the, "...vast quantity of data that is related to scholarly undertaking" [25], much of which is available on the World Wide Web. It is estimated that there are at least 114 million English scholarly documents or their records [1] accessible on the Web [14].

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 autononmously, even leveraging utility-based feedback control to minimize computational resource usage and incorporate user input to correct automatically extracted metadata [26]. 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 fuctionalities, 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: Context & Architecture

While major engines, such as Microsoft Academic Search and Google Scholar, and online digital repositories, such as DBLP, provide publication and bibliographic collections 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 exempified through various derived data-sets, ranging from citation graphs to publication acknowledgements [15], 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 [31] to meta-data extraction to leveraging user-provided meta-data corrections [27]. 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.

CiteSeerX can be compactly described as a 3-layer complex system (as shown in Figure 1 [2]). 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

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

[2] Note: This overview displays the proposed, private cloud-based CiteSeerX platform described in [31], though several modules in this paper exist in the current CiteSeerX system or are under active development.

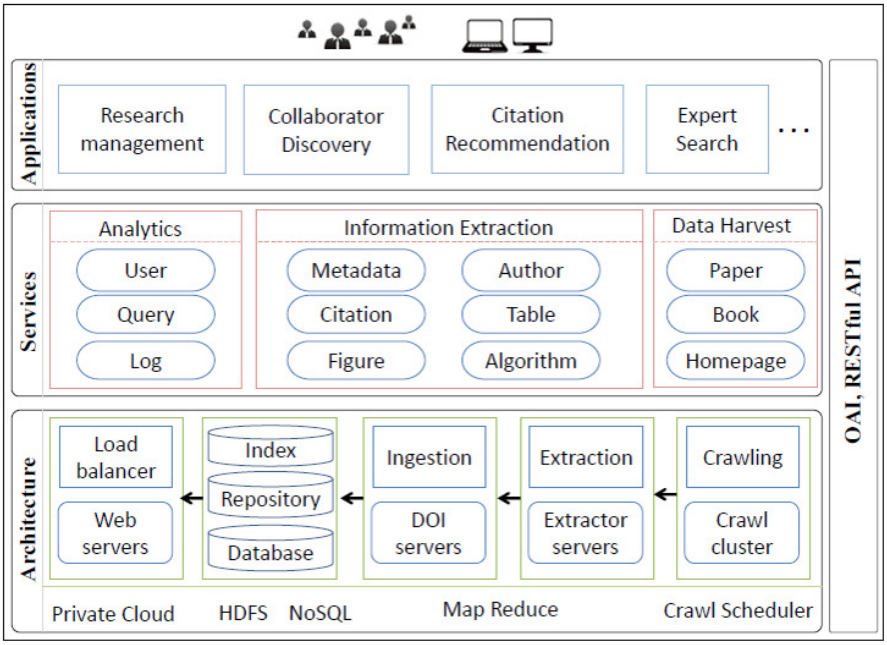

Figure 1: High-level overview of the CiteSeerX platform.

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 [3], 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 frontline for information gathering. Some agents are used in scheduled re-crawls using known seed URLs to improve the freshness of CiteSeerX's data while others are pointed to begin crawling newly discovered locations (some provided by users). Currently, a rule-based filter is used to determine if documents crawled by these agents are academic or not [4]. 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.

[3] For more detailed explications of the CiteSeerX architecture itself, we refer the reader to [31, 27].

[4] 80–90% precision and 70–80% recall on a manually labelled document set

While our discriminative model is designed to better filter document data harvested by CiteSeerX'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 [11]. Such a functionality can be achieved by automatic classification of web pages based on both the URL and crawl history.

4 Information Extraction

4.1 Header Extraction

Headers, which contain useful information fields such as paper title and author names, are extracted using SVMHeaderParse [12], 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 variables. 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 [19] 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).

4.2 Extracting Citations

CiteSeerX uses ParsCit [10] for citation extraction, which has a conditional random field model core [16] 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.

While ParsCit performs reference string segmentation reasonably well, it makes some mistakes in segmentation as it 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.

4.3 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.

4.4 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 [21].

4.5 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 25% of CiteSeerX papers have matching counterparts in DBLP with 80% recall and 75% precision. 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.

5 Looking to the Future: Facilitating Knowledge Generation

5.1 Algorithm Search

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 pseudocode [22]. Three methods for detecting pseudocode in scholarly documents include rule based, machine learning based, and combined methods. We found that combined methods perform the best (F1 score) for extracting indexable metadata (captions, textual summaries) for each detected pseu-

docode. On the other hand, extracting algorithm-specific metadata (such as algorithm name, target problems, or runtime 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 and ranking. We have begun exploring the mining of algorithm semantics by studying the algorithm co-citation network [23] 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 [24], which would then allow for automatic construction of a large scale algorithm citation network.

5.2 Figure Search

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 used 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 [20]. We focused on analyzing line graphs, given their highly frequent usage in research papers to report results. Following the work of [20], 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 facilitate information extraction and knowledge creation. 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, to mine 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 nature of academia. These include *RefSeer* for recommending topic and context-related citations given a portion of a paper [13], *CollabSeer* for discovering potential collaborators for a given author [5], and *CSSeer*, a Computer Science expert discovery and related topic recommendation system [6].

Other aspects of scholarly data that CiteSeerXhandles include tables [18], acknowledgements [15], table of contents [30], and back-of-the-book indices [29, 28]. 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 CiteSeerXsystem 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.

References

[1] BHATIA, S., CARAGEA, C., CHEN, H.-H., WU, J., TREERATPITUK, P., WU, Z., KHABSA, M., MITRA, P., AND GILES, C. L. Specialized research datasets in the CiteSeerX digital library. In *D-Lib Magazine* (2012), vol. 18.

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

[3] CARAGEA, C., WU, J., CIOBANU, A., WILLIAMS, K., FERNANDEZ-RAMIREZ, J., CHEN, H.-H., WU, Z., AND GILES, C. L. Citeseerx: A scholarly big dataset. ECIR '14, pp. 311–322.

[4] CHAO, H., AND FAN, J. Layout and content extraction for pdf documents. In *Document Analysis Systems VI*. Springer, 2004, pp. 213–224.

[5] CHEN, H.-H., GOU, L., ZHANG, X., AND GILES, C. L. Collabseer: a search engine for collaboration discovery. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries* (2011), ACM, pp. 231–240.

[6] CHEN, H.-H., TREERATPITUK, P., MITRA, P., AND GILES, C. L. CSSeer: An expert recommendation system based on CiteseerX. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries* (2013), JCDL '13, ACM, pp. 381–382.

[7] CHOUDHURY, S. R., MITRA, P., KIRK, A., SZEP, S., PELLEGRINO, D., JONES, S., AND GILES, C. L. Figure metadata extraction from digital documents. In *Proceedings of ICDAR* (2013), IEEE, pp. 135–139.

[8] CORNELIA CARAGEA, JIAN WU, K. W. S. D. G. M. K. P. T., AND GILES., C. L. Automatic identification of research articles from crawled documents. In *Proceedings of WSDM-WSCBD* (2014).

[9] CORTES, C., AND VAPNIK, V. Support-vector networks. *Machine Learning 20*, 3 (1995), 273–297.

[10] COUNCILL, I. G., GILES, C. L., AND KAN, M.-Y. Parscit: an open-source crf reference string parsing package. LREC '08.

[11] GOLLAPALLI, S. D., GILES, C. L., MITRA, P., AND CARAGEA, C. On identifying academic homepages for digital libraries. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries* (New York, NY, USA, 2011), JCDL '11, ACM, pp. 123–132.

[12] HAN, H., GILES, C. L., MANAVOGLU, E., ZHA, H., ZHANG, Z., AND FOX, E. A. Automatic document metadata extraction using support vector machines. In *Digital Libraries, 2003. Proceedings. 2003 Joint Conference on* (2003), IEEE, pp. 37–48.

[13] HUANG, W., KATARIA, S., CARAGEA, C., MITRA, P., GILES, C. L., AND ROKACH, L. Recommending citations: translating papers into references. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (2012), ACM, pp. 1910–1914.

[14] KHABSA, M., AND GILES, C. L. The number of scholarly documents on the public web. *PloS one 9*, 5 (2014), e93949.

[15] KHABSA, M., TREERATPITUK, P., AND GILES, C. L. 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* (2012), ACM, pp. 185–194.

[16] LAFFERTY, J. D., MCCALLUM, A., AND PEREIRA, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ICML '01, pp. 282–289.

[17] LIPINSKI, M., YAO, K., BREITINGER, C., BEEL, J., AND GIPP, B. 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* (New York, NY, USA, 2013), JCDL '13, ACM, pp. 385–386.

[18] LIU, Y., BAI, K., MITRA, P., AND GILES, C. L. TableSeer: Automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries* (2007), JCDL '07, ACM, pp. 91–100.

[19] LOPEZ, P. 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* (Berlin, Heidelberg, 2009), ECDL'09, Springer-Verlag, pp. 473–474.

[20] LU, X., KATARIA, S., BROUWER, W. J., WANG, J. Z., MITRA, P., AND GILES, C. L. Automated analysis of images in documents for intelligent document search. *IJDAR 12*, 2 (2009), 65–81.

[21] TREERATPITUK, P., AND GILES, C. L. Disambiguating authors in academic publications using random forests. JCDL '09, pp. 39–48.

[22] TUAROB, S., BHATIA, S., MITRA, P., AND GILES, C. Automatic detection of pseudocodes in scholarly documents using machine learning. In *Proceedings of ICDAR* (2013).

[23] TUAROB, S., MITRA, P., AND GILES, C. L. Improving algorithm search using the algorithm co-citation network. In *Proceedings of JCDL* (2012), pp. 277–280.

[24] TUAROB, S., MITRA, P., AND GILES, C. L. A classification scheme for algorithm citation function in scholarly works. In *Proceedings of JCDL* (2013), JCDL '13, pp. 367–368.

[25] WILLIAMS, K., WU, J., CHOUDHURY, S. R., KHABSA, M., AND GILES, C. L. Scholarly big data information extraction and integration in the CiteSeer digital library. In *Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on* (2014b), IEEE, pp. 68–73.

[26] WU, J., ORORBIA, A., WILLIAMS, K., KHABSA, M., WU, Z., AND GILES, C. L. Utility-based control feedback in a digital library search engine: Cases in CiteSeerX. In *9th International Workshop on Feedback Computing (Feedback Computing 14)* (2014), USENIX Association.

[27] WU, J., WILLIAMS, K., CHEN, H.-H., KHABSA, M., CARAGEA, C., ORORBIA, A., JORDAN, D., AND GILES, C. L. Citeseerx: Ai in a digital library search engine. In *The Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence* (2014), IAAI '14.

[28] WU, Z., AND GILES, C. L. Measuring term informativeness in context. In *Proceedings of NAACL-HLT 2013* (2013), p. 259269.

[29] WU, Z., LI, Z., MITRA, P., AND GILES, C. L. Can back-of-the-book indexes be automatically created? In *Proceedings of CIKM* (2013), pp. 1745–1750.

[30] WU, Z., MITRA, P., AND GILES, C. Table of contents recognition and extraction for heterogeneous book documents. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)* (2013), pp. 1205–1209.

[31] WU, Z., WU, J., KHABSA, M., WILLIAMS, K., CHEN, H.-H., HUANG, W., TUAROB, S., CHOUDHURY, S. R., ORORBIA, A., MITRA, P., AND OTHERS. Towards building a scholarly big data platform: Challenges, lessons and opportunities. In *Proceedings of the International Conference on Digital Libraries 2014* (2014), vol. 447, JCDL 2014, p. 12.