ABSTRACT

We overview CiteSeerX, the pioneer digital library search engine, that has been serving academic communities for more than 20 years (first released in 1998), from three perspectives. The system perspective summarizes its architecture evolution in three phases over the past 20 years. The data perspective describes how CiteSeerX has created searchable scholarly big datasets and made them freely available for multiple purposes. In order to be scalable and effective, AI technologies are employed in all essential modules. To effectively train these models, a sufficient amount of data has been labeled, which can then be reused for training future models. Finally, we discuss the future of CiteSeerX. Our ongoing work is to make CiteSeerX more sustainable. To this end, we are working to ingest all open access scholarly papers, estimated to be 30-40 million. Part of the plan is to discover dataset mentions and metadata in scholarly articles and make them more accessible via search interfaces. Users will have more opportunities to explore and trace datasets that can be reused and discover other datasets for new research projects. We summarize what was learned to make a similar system more sustainable and useful.

CCS CONCEPTS

• Information systems → Digital libraries and archives; Information integration; • Computing methodologies → Information extraction.

KEYWORDS

CiteSeerX, digital libraries, search engines, scholarly big data

1 INTRODUCTION

The number of scientific publications has been increasing exponentially after the mid 1900’s [14]. This poses a great challenge in managing a large number of documents and providing timely access to a growing number of researchers. Mass digitization partially solved the problem by storing document collections in digital repositories. The advent of modern information retrieval methods significantly expedited the process of relevant search. However, documents are still saved individually by many users. In 1997, three computer scientists at the NEC Research Institute (now NEC Labs), New Jersey, United States – Steven Lawrence, Kurt Bollacker, and C. Lee Giles, conceived an idea to create a network of computer science research papers through citations, which was to be implemented by a search engine, the prototype CiteSeer. Their intuitive idea, automated citation indexing [8], changed the way researchers searched for papers. Users were readily able to navigate from one paper to another by tracking citation relationships.

CiteSeer first served the academic community in 1998 (mostly computer science). It is usually recognized as the first digital library search engine (DLSE)\(^1\). In 2008, CiteSeer was renamed CiteSeerX, where “X” stands for a series of enhancements as well as architecture and infrastructure redesigns. As a production system based in an academic setting, CiteSeerX has been steadily growing. A relatively small team overcame many scientific and technical challenges with the goal of making the system more accurate, accessible, and scalable. The current team has designed and implemented algorithms for several outstanding problems such as citation parsing [5], table extraction, e.g., [16], author name disambiguation, e.g., [21], document classification, e.g., [3], and data cleansing, e.g., [18].

Although there are similar DLSEs available nowadays. CiteSeerX maintains a unique position. (1) It uses a focused web crawler to actively crawl the public Web. This is different from ACM DL, and IEEE Xplore, where the metadata is entered by authors and provided by these publishers. Google Scholar obtains its data from both publishers and the Web and redirects users to webpages containing documents that are not necessarily open access (OA). Microsoft Academic’s data is released via the Academic Knowledge API behind a pay wall. Unlike Academia.edu, ResearchGate, and AMiner, CiteSeerX does not solicit paper uploads from individual authors. CiteSeerX is an OA digital library and users have access to full-text of all documents searchable on its website. All papers are associated with public URLs. (2) CiteSeerX provides all automatically extracted metadata and citation context via an OAI (Open Archive Initiative) interface. The data can also be downloaded from a publicly available drive under a Creative Commons (CC) license, a service not available from Google Scholar and Semantic Scholar. (3) Nearly all CiteSeerX papers are indexed by Google Scholar. (4) CiteSeerX provides an open source software framework called SeerSuite, which has been deployed at other sites such as the Qatar University Library.

\(^1\)For reference, Google Scholar was launched in 2004; Windows Live Academic Search, later renamed Microsoft Academic Search, now called Microsoft Academic was launched in 2006; Semantic Scholar was launched in 2015.
The evolution of the CiteSeer System can be divided into three phases: the single machine phase (1997–2003), the multiple server phase (2003–2013), and the private cloud phase (2013–present). The frontend designs have changed significantly over the years, as seen in Figure 1.


The original CiteSeer at NEC Research Institute was developed and deployed on a single server. The web service was based on Apache HTTP. Because there was not many open source software packages that fit their goals, the developers wrote almost all software by themselves in Perl for the web crawler, the indexer, and the search API. The search engine also used the name ResearchIndex at one point. The crawlers were seeded from manually curated homepage URLs of computer scientists. The search engine indexed about 220,000 documents with 2.5 million citations.

### 2.2 Phase II: Physical Cluster (2003–2013)

In 2003, an NSF SGER grant allowed CiteSeer to be moved to the College of Information Sciences and Technology (IST) at the Pennsylvania State University (PSU). To overcome the capacity limit of a single machine, the search engine evolved to a multi-server system. Before 2007, there were 8 servers including 2 load balancers, 2 web servers, 3 repository servers, and 1 staging server (for development and web crawling). In 2005, the NSF CRI grant proposed to develop the next generation of CiteSeer and scaled up the system to 14 servers. CiteSeer was renamed to CiteSeerX in 2008. Lucene was introduced as the main indexer. In 2011, Apache Solr was adopted as the main indexer, with about 2 million academic documents.

The backbone software was rewritten using Java [6], which used a digital library framework SeerSuite [20]. The web application uses a model-view-controller architecture implemented with the Spring framework. The frontend uses a mix of Java server pages and JavaScript to generate user interfaces. The web application is composed of servlets that interact with the index and database for keyword search and uses Data Access Objects to interact with databases and the repository. The entire data is partitioned across 3 major databases: user information, document metadata, and citation graphs. The metadata extraction method was built in Perl, working in batch mode. The ingestion system, which feeds the database and repository was integrated into the web application. The data were acquired using an incremental web crawler developed using Django. The crawler discovered 700k+ parent URLs linking to OA PDFs by 2013.

### 2.3 Phase III: Private Cloud (2013–present)

A thorough analysis indicated a private cloud was the most economic and efficient way to overcome the bottlenecks of system maintenance and scalability [27]. In 2013, supported by an NSF grant CiteSeerX was successfully migrated into a private cloud. The infrastructure consists of 3 layers. The storage layer includes 2 servers for storing virtual machines (VMs); the processing layer includes 5 high-end servers running VMs for web service, database, etc.; the software on VMs runs on the application layer. At least 20 VMs are created and running in the private cloud. The web crawler, due to its high demand on bandwidth and disk access, is hosted on a physical server. The software was basically inherited from Phase II, but the system was enhanced with features such as author and table searches, built on author name disambiguation, e.g., [21], table extraction, e.g., [16], etc.

### 2.4 Usage and Community Benefit

By 2017, CiteSeerX had ingested the metadata and full text of more than 10 million OA academic documents on the Web and it is increasing. According to Google Analytics and local access logs, CiteSeerX has almost 3M individual users in Year 2017 and has 500,000 documents downloaded daily with on average 3 million hits per day. The OAI is accessed approximately 5000 times monthly [23]. A Google search of “CiteSeerX OR CiteSeer” returns about 10M results. CiteSeerX has a world-wide user population. The top 5 countries in 2017 were China (33%), United States (27%), India (11%), United Kingdom (7%), and Germany (7%). Access log analysis in 2015 indicated that approximately 5000 and 7000 accesses per day are from Historical Black Colleges and Universities and Hispanic Serving Institutes, respectively.

### 3 AI AND REUSABLE DATA

#### 3.1 AI in CiteSeerX

CiteSeerX incorporates AI technologies in many mission critical tasks. Figure 2 illustrates the procedures for mining and extracting scholarly big data in CiteSeerX, starting from raw PDFs to various data and AI-based software products. The PDFs are first classified into academic and non-academic documents. Machine learning based classifiers in place of the rule-based classifier boosts the $F_1$ by at least 10% [3]. For information extraction (IE), SVMHeaderParse is replaced [9] with GROBID [17] because of performance [15]. For non-textual information extraction, we use pdf2figures2 [4] to extract figures and tables. Algorithms were also developed to extract algorithms [22] and chemical entities [12]. Math expression extraction is still under active research, e.g., [30]. We perform document deduplication (conflation), keyphrase extraction [2], and author
name disambiguation [21]. We also link local databases with external databases [26]. Finally, a knowledge base can be constructed using typed entities [24] and relations extracted from body text, plus semantics extracted from non-textual information [1].

It should be noted that there have been a myriad of AI-based algorithms proposed for tasks above, but many cannot be adopted mainly due to 3 reasons: (1) published work had results were not repeatable; (2) the algorithm could not be scaled to big data; (3) the work was based on a toy model with unrealistic assumptions of input data quality. As such we strive to develop portable and scalable AI-based software and adopt new algorithms and implementations that can improve data quality, quantity, and user experience. Algorithms that require significant transfer overhead or not designed for big data are usually inappropriate.

3.2 Reusable Data

CiteSeerX offers two types of reusable data – automatically extracted data (AED) and manually labeled data (MLD). Types of AED and their sizes are tabulated in Table 1. papers to be indexed are obtained by IE (Figure 2). The disambiguated authors are obtained using random forests and DBSCAN clustering [7]. Citations and citation context are extracted using ParsCit [5]. The citation graph is generated heuristically [28]. Data in Table 1 are stored in MySQL and dumped into a .sql file. It takes about 550GB after imported into the database. The unique bibliographic records indexed by Solr is 360GB. The repository containing all types of files (PDF, XML, TXT, etc.) takes 15TB.

Table 2 presents examples of MLDs. These datasets can be reused for training and evaluating new models. Two examples below demonstrate how these datasets can be used.

For the first example, we attempt to cleanse the metadata produced by IE [18]. Data quality is a ubiquitous problem for automatic extraction pipelines. The errors in metadata can propagate and lead to unreliable results in downstream analysis. One approach is to cleanse the dataset (called target dataset) by matching it against a clean reference dataset, and then use reference data to overwrite target data. To train such a model, we developed the paper entity matching dataset (Table 2) containing 688 matching pairs between CiteSeerX and reference databases (DBLP, IEEE, etc.), with an equal amount of negative matching pairs. By matching headers alone, the model achieves $F_1 \sim 92\%$. By matching headers and citations, the model achieves $F_1 > 99\%$. The dataset CiteSeerX-2018 (Table 1) comes from matching the entire CiteSeerX database with DBLP, and Medline [25].

Another example is disambiguating author mentions in academic papers. Name disambiguation is a common and important issue (unfortunately often ignored) for nearly all problems involving author names. The goal is to build a model that cluster the same surface name corresponding to different individuals (e.g., Michael Jordan, a computer scientist or a basketball player?) and different surface names referring to the same individual (e.g., CL Giles, Lee Giles, and C. Lee Giles all are the same person). The author name disambiguation dataset was constructed in 2004 [10] and has been reused in several papers for the same task [11, 13, 19, 21, 29].

4 LESSONS LEARNED

As a system designed to serve the academic and research community, CiteSeerX is one of the few systems that still exists after 20 years. To keep the system up to date, the CiteSeerX team undertakes both scientific research and system design and development. As such there are lessons learned that may benefit related systems of similar size and functionality. (1) Maintenance is extremely important. Several other very good systems did not last long because

![Figure 2: Overview of the mining of scholarly big data in CiteSeerX. Green boxes are tasks that use AI-based software.](image_url)
of poor maintenance, such as coding, lack of documentation, system frailty, and IT support. (2) Research and the system need to be strongly coupled. Research provides cutting-edge tech support for the system while the system provides real data and test beds for research. (3) The system must provide a reliable and unique service to maintain a considerable user population. For CiteSeerX, this means complete open source software (where possible) and data including all documents.

5 FUTURE PROSPECTS

The future of CiteSeerX relies on both research and system development. Metadata extraction has made great progress in the past decade, but much of the text is still relatively unexplored. For example, although there are OA data repositories (e.g., figshare) and search engines (e.g., Google Data Search), a myriad of datasets mentioned in academic papers that are not discovered or used. We propose research on AI-based algorithms that extract datasets and their associated metadata from the full text of academic papers in multiple domains. Two of the biggest challenges are the lack of domain knowledge and the large amount of training data. In CiteSeerX, papers in different scientific domains are all together. It would be useful to first classify them by subject categories. In a preliminary study, we used feature-based machine learning models and a multilayer perceptron to classify papers into 6 categories and achieved a micro-$F_1 \approx 0.83$ [25]. We are also experimenting with deep neural networks in order to expand the current method to 104 scientific categories. To build the labeled corpus, we will first build author profiles and then request annotations from authors and readers.

From a sustainability perspective, we are investigating a four-sided model that will sustain CiteSeerX for the next 10 years. This includes (1) increasing coverage and freshness of the collection, (2) improving metadata quality with state-of-the-art extractors and data cleansing modules, (3) employing ElasticSearch as an indexer and a metadata storage in place of a database; and (4) enriching semantic data extracted from full text.

ACKNOWLEDGMENTS

We gratefully acknowledge partial support from National Science Foundation and thank the reviewers for constructive comments.

REFERENCES