ABSTRACT
CiteSeerX is a digital library search engine that provides free access to over six million scholarly documents crawled from the public web. Their metadata is automatically extracted and tagged. We present key extraction technologies used in CiteSeerX, including document classification and de-duplication, document clustering, header/citation extraction, author disambiguation, and table/algorithm extraction. We also describe developing challenges and future work.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous; I.2.1 [Artificial Intelligence]: Applications and Expert Systems

1. INTRODUCTION
CiteSeerX's predecessor, CiteSeer, was developed at the NEC Research Institute, Princeton, NJ in 1997 and was considered by many to be the first scholarly digital library that provided autonomous citation indexing [15]. At Penn State since 2003, CiteSeer was renamed as CiteSeerX in 2008 with a new architecture and features and continued to be heavily used. A future goal is to utilize the metadata for various types of semantic search.

CiteSeerX is in many ways unique compared with other scholarly digital libraries and search engines since all documents are harvested from the public Web. Because of this, users have full-text access to all papers searchable in CiteSeerX. Also, CiteSeerX performs automatic extraction and indexing on paper entities such as tables and figures, which is rarely seen in other scholarly search engines. The overall accuracy of this extractor is 92.9%, which is better than the accuracy (90%) reported by [32]. Recent evidence implies that GROBID [28] would be a good replacement [25].

2. EXTRACTION TECHNOLOGIES

Document Classification. Text content extracted from the PDF of crawled documents is filtered to determine whether the document is scholarly or not. A more sophisticated machine learning approach [3] utilizes structural features to classify documents, including File specific features, Text specific features, Section specific features, and Containment features. These new classifiers significantly outperform our previous baselines in terms of precision, recall, and accuracy by at least 10%.

Header Extraction. Header extraction is performed using a support vector machine parser, SVMHeaderParse [17] based on svm-light [19]. It classifies textual contents into multiple classes, each of which corresponds to a header metadata field, e.g., title, authors. The overall accuracy of this extractor is 92.9%, which is better than the accuracy (90%) reported by [32]. Recent evidence implies that GROBID [28] would be a good replacement [25].

Citation Extraction. CiteSeerX uses ParsCit [13] for citation extraction, which is a conditional random field (CRF; Lafferty et al. 2001) model that labels the token sequences in reference strings. ParsCit first attempts to find the reference section before parsing reference strings and then searches for where the individual reference starts and ends using either reference markers or heuristic methods. It also extracts citation context. Evaluations show that the performance of ParsCit is comparable to the original CRF based system in Peng & McCallum (2004), and outperforms FLUX-CiM [12].

De-duplication. Near-duplicates (NDs) refer to documents with similar content but minor differences. NDs are very common in crawl-based digital libraries. In CiteSeerX, NDs are detected using a key-mapping algorithm, applied after the metadata extraction module but before papers are ingested. When a document is imported, a set of keys is generated by concatenating normalized author last names and normalized titles. The key-mapping algorithm [38] is comparable to the state-of-the-art simhash approach [4].
**Author Disambiguation.** CiteSeerX provides a special author search interface. Author search is also the foundation of several other services, such as collaborator search [5] and expert search [6]. Processing a name-based query can be complex since different authors may share the same name and the same author may have several name variations. To disambiguate authors, we block names into small blocks with the assumption that an author can only have different name variations within the same block. CiteSeerX groups two names if they share the same last name and the first initials. In many cases, other information related to authors is used including their collaborators and topics of their published papers. Our algorithm applies DBSCAN (Density-Based Spacial Clustering of Application with Noise) to resolve most of inconsistent classification results violating a transitivity principle [18]. The Random Forest training of the distance function [36] scales well and has decent performance [22].

**Table Extraction.** CiteSeerX uses the table metadata extractor developed by Liu et al. (2007), which is comprised of three major parts: a text information stripper, a table box detector, and a table metadata extractor. The text information stripper extracts out the textual information from the original PDF files word by word by analyzing the output of a general text extractor. These words are then reconstructed with their position information and written into a document content file, which specifies the position, line width and fonts of each line. Based on the document content file, the tables are identified using a box-cutting method, which attempts to divide all literal components in a page into “boxes”. Finally, the algorithm finds tables and their metadata in these boxes [27].

**Algorithm Extraction.** We developed three methods for detecting pseudo-codes in scholarly documents based on textual content extracted from PDF documents. The rule-based method detects the presence of pseudo-code captions using a set of regular expressions. This method yields high detection precision (87%), but low recall (45%), because a large proportion of pseudo-codes (roughly 26%) do not have associated captions. A machine-learning method directly detects the presence of pseudo-code content assuming that pseudo-codes are written in a sparse, programming-like manner, which can be visually spotted as sparse regions in documents and can capture most non-captioned pseudo-codes. The ones that cannot be captured are either written in a descriptive manner or are presented as figures. A hybrid method combines both and achieves a precision of 87% and a recall of 67% [37].

3. **USAGE AND PAYOFF**

Since 2008, the document collection of CiteSeerX has been steadily growing, now at six million. Currently, CiteSeerX servers are hit more than 2 million times a day and 3–10 PDF files are downloaded per second [34]. Besides the web search, CiteSeerX also provides an OAI protocol for metadata harvesting in order to facilitate content dissemination [40]. Dumps of our database are also available on Amazon S31. CiteSeerX data is updated regularly. The crawler downloads 50,000 to 100,000 PDF files per day, and up to 50,000 new papers are ingested every day. The CiteSeerX data is heavily used in research projects, e.g., [14, 29, 31, 2, 1]. CiteSeerX has released the open source digital library search engine framework, SeerSuite [35], which can be used for building personalized digital library search engines.

4. **NEW EXTRACTION FRAMEWORK**

In general, a scholarly document consists of several of these entities, if not all: a header, a text body, a bibliography, figures, tables, math and algorithms (even chemical formulae [33]). Recently, we developed a multi-entity knowledge extraction framework for scholarly documents in PDF format called PDFMEF [41]2. It is implemented with a framework that encapsulates open-source extraction tools. Currently, it leverages PDFBox and TET for full text extraction, the scholarly document filter introduced in [3] for document classification, GROBID for header extraction, ParsCit for citation extraction, PDFFigures [11] for figure and table extraction, and algorithm extraction algorithm introduced in [37]. While it can be run out-of-box, the extraction tool in each module is customizable. The framework is designed to be scalable and is capable of running in parallel using a multi-processing technique in Python.

5. **CHALLENGES AND FUTURE WORK**

Two big challenges in CiteSeerX are data acquisition and information quality. Previously, the majority of CiteSeerX papers were from the computer sciences. Recently, a large number of papers have been collected from mathematics, physics, and medical science by incorporating papers from open-access repositories such as PubMed (subset), and crawling URLs released by Microsoft Academic Search. Our experiments indicate that the crawl efficiency increases by at least 20% using a whitelist policy [42]. One extension of the crawl module is to integrate a crawl scheduler that generates whitelists on a daily basis, which used the webpage updating rate as a selection criteria based on estimated crawl history [7, 9]. To increase coverage and freshness, the crawling process should be parallelized [8]. In the near future we hope to harvest the estimated 25 million freely available scholarly documents [21] on the web.

To increase metadata quality, we recently developed PDFMEF, which can be used to rebuild the entire metadata database. We are also using multiple manually created reference data sets, such as from DBLP, some publishers, etc., to sanitize and correct mistakenly extracted metadata. We have also developed a multi-document-type classifier, which classifies crawled documents into finer categories, such as slides, papers, and theses. This classifier will be used to build a large data corpora for information extraction research. We intend to link multiple documents types to scholarly documents and make them accessible from a federated view. New features that could be incorporated into CiteSeerX are algorithm search [37], figure search [10], and acknowledgment search [16, 23].

6. **CONCLUSION**

CiteSeerX is an open access digital library search engine which has incorporated multiple information extraction technologies and plans to grow much larger and improve its

1Accessible upon request.

2https://github.com/SeerLabs/new-csx-extractor
metadata quality. In addition, other metadata such as chemical formulae can be extracted, linked, and used for other types of search including semantic search.

7. ACKNOWLEDGMENTS

We gratefully acknowledge partial support from the National Science Foundation.

8. REFERENCES


