CiteSeerX Data: Semanticizing Scholarly Papers

Jian Wu[†], Chen Liang[†], Huaiyu Yang[*], C. Lee Giles[†‡]
[†]Information Sciences and Technology
[‡]Computer Science and Engineering
Pennsylvania State University, University Park, PA, 16802 USA
[*]Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, 37235

ABSTRACT

Scholarly big data is, for many, an important instance of Big Data. Digital library search engines have been built to acquire, extract, and ingest large volumes of scholarly papers. This paper provides an overview of the scholarly big data released by CiteSeerX, as of the end of 2015, and discusses various aspects such as how the data is acquired, its size, general quality, data management, and accessibility. Preliminary results on extracting semantic entities from body text of scholarly papers with Wikifier show biases towards general terms appearing in Wikipedia and against domain specific terms. We argue that the latter will play a more important role in extracting important facts from scholarly papers.

CCS Concepts

•**Applied computing** → **Digital libraries and archives;** •**Information systems** → *Extraction, transformation and loading; Deduplication;* •**Computing methodologies** → Lexical semantics;

Keywords

CiteSeerX, Digital Library Search Engine, Scholarly Big Data, Citation Graph, Semantic Entity Extraction

1. INTRODUCTION

Big Data can be classified into several categories depending on the source, e.g., data from sensors, social interactions, business interactions, electronic files, and broadcastings. The scholarly data in this paper refers that extracted from scholarly papers as electronic files. Characteristics of Big Data are often described with three dimensions, "Volume", "Variety", and "Velocity". Scholarly data has a large volume; it was estimated in 2014 that the total number of scholarly papers on the Web was about 120 million [11], and about a quarter were freely accessible. Assuming that most are in PDF and the average size of a scholarly paper is 1 MB, then all papers would be 120 TB and 30 TB for those freely accessible. For comparison, this size is larger than the NASA Earth Exchange Downscaled Climate Projections dataset (17 TB) available on AWS. Scholarly publications are growing at a rate of over 1 million annually [13].

There are various types of scholarly data. Document text is mainly unstructured while the paper as a whole, separated by section identifiers, and itemizations, is semi-structured. Metadata extracted and parsed from papers is structured. Finally, while the volume of academic documents continues to grow, digital library search engines are expected to respond to search queries on a sub-second time scale. Scholarly data is also very rich in facts and knowledge. As such scholarly data can be considered an important instance of big data.

Digital library search engines (DLSEs) are commonly used to manage and search scholarly big data, either through Web UIs, APIs, or digital copies. Examples of crawl-based DLSEs include Google Scholar, Microsoft Academic Search, Semantic Scholar, and CiteSeerX. These search engines acquire their documents by the most part by actively crawling the Web. Information and metadata are automatically extracted and indexed. Submission-based DLSEs, instead, receive metadata directly from publishers, or from manual input, such as Harvard ADS, PubMed, Web of Science, Elsevier, and arXiv. This data is typically focused on particular fields, such as astrophysics (Harvard ADS). Due to copyright, a large fraction of full papers in some of these DLSEs is not public. As a result, crawl-based DLSEs are important sources of research data for tasks such as citation recommendation, e.g., [10, 4], author name disambiguation e.g., [25, 12], ontologies, e.g., [1], document classification, e.g., [3], and "Science of Science", e.g., [21].

One of the limitations of many digital library data releases is that they do not include full text and author name disambiguation. For example, the DBLP data [15] contains metadata of 4.8 million papers (as of 2015), but it only includes header metadata (e.g., title, authors, year, and venue). The recently released Microsoft Academic Graph [22] contains over 100 million paper records (2015-11-06), which is the largest academic publication metadata release so far but author names are not disambiguated (which is non-trivial), and the full text is not available. There are other scholarly paper providers such as CORE and OpenDOAR that use an approach called "harvesting". Different from "web crawling", these digital libraries passively receive data or metadata from open access repositories, which could lead

SBD'16, July 01 2016, San Francisco, CA, USA

ISBN 978-1-4503-4299-5/16/07 . . . $15.00
DOI: http://dx.doi.org/10.1145/2928294.2928306

to a strong bias in document content. Empirically, not all papers have full text, and data access is usually limited to a web interface.

The CiteSeerX data is in some cases unique compared with the data sources above. The data release contains full text of more than 7 million (as of the beginning of 2016) open access scholarly documents and author metadata has been disambiguated for the main database. For this data we present preliminary results for semanticizing scholarly papers, as an application on the data, and as an effort towards building a scholarly knowledge base.

2. ACQUISITION AND EXTRACTION

The data in CiteSeerX is collected from two sources. A focused web crawler actively harvests open access documents from the Web. The first set of seed URLs were manually curated HTML pages, mostly homepages of professors and researchers in computer science. The crawler downloads PDF documents linked to these pages, finds new URLs from these pages, and saves useful parent URLs into the crawl database. In this way, a large collection of PDF files form the crawl repository. Each PDF file is associated with its original URL, and a parent URL (if any), which can be re-crawled for updates. Since 2012, we started to crawl a whitelist containing high quality URLs selected from all parent URLs [28]. Meanwhile, Heritrix, the web crawler of Internet Archive, replaced our own crawler. Heritrx implements a sophisticated thread pool manager that balances politeness and aggressiveness. We also host a crawl website for individual users and publishers to submit URLs and have received over 200,000 user URLs since 2009. Seed URLs are also adopted from public data releases such as Wikipedia External Links and Microsoft Academic Graph. Given current limitations, we can crawl up to 200,000 PDF files per day using a single server. The crawler strictly obeys `robots.txt`. Documents are also downloaded directly from open access repositories, such as PubMed Central, and arXiv. We automatically construct URLs that link to the original pages of these papers.

After files are crawled, they are labeled with a crawl ID and imported into the crawl repository. The associated metadata, including the crawl time, original URLs, parent URLs, URL hash, and content hash are saved into a crawl database. The crawled documents are then processed by the extraction module in batches. At first, the full text is extracted. Before 2015, we used PDFLib TET 3.0. Since 2015, we have used Apache PDFBox 1.8.4, an open source toolkit with comparable performance to TET 4.0 [27]. A rule-based filter is applied on each document [29]. Academic documents are passed to metadata extraction. Header metadata are extracted with SVMHeaderParse [9], including title, authors, date (if available), and abstract. Also extracted is name, affiliation, and address for each author. ParsCit [6] is used to extract and parse references. For each reference, ParsCit extracts authors, title, venue, venue type, year, book title, location, journal, date, volume, pages, note, marker, raw string, as well as citation context, which is the text around where the reference is cited. After citation extraction, an XML file is generated to concatenate header and citation extraction results. The ingestion process reads all fields in the XML file and inserts their values into the production database. Near-duplicated documents (NDs) are ubiquitous. These documents usually have the same (or similar) titles and author lists but different checksums. To identify NDs, the ingestion module generates string keys from each document with title and author names, and groups papers with one or more common string keys into a *document cluster* with an ID (called CID). A new document is ingested with a unique ID (called csxdoi), and is also assigned a CID. Citation ingestion is handled in a similar way. Therefore, a document cluster may contain documents or citations or both. At the end of each ingestion cycle, a *new* XML file is generated containing the revised metadata (clustering and inference). This XML file, along with the original PDF, and auxiliary files are copied into the production repository. Currently, we ingest 20,000 to 30,000 documents per day with a peak around 60,000.

3. DATA PRODUCTS

3.1 Raw Data

CiteSeerX offers two levels of data: raw and processed. The raw data includes the crawl repository and database. The size of the crawl repository has increased remarkably since 2008 (Table 1). The crawl database contains two major tables: `main_crawl_document` (26 million rows) and `main_crawl_parenturl` (2.5 million rows). While the crawl database takes only 16 GB of space, the crawl repository takes over 24 TB of space. Duplicate elimination is handled such that a document is considered new as long as the content hash *and* the URL hash do not match existing record. Then the document is imported. This preserves multiple sources linking to the same document, as well as possible updates of a document from the same URL. Crawl statistics [1] have country ranking by number of documents, domain (or top level domain) rankings by number of documents, citations, and citation number per document. Users can also submit URLs to be crawled from this portal.

Table 1: Document collection since 2008 (two-digit year). Indexed documents represent unique ones among all the ingested. Numbers are in millions.

Year	08	09	10	11	12	13	14	15
Crawled	1.9	2.9	5.6	6.2	7.9	13.0	21.0	25.8
Ingested	0.6	1.4	1.7	1.9	2.4	3.8	5.1	6.9
Indexed	0.5	0.8	1.0	1.2	1.5	2.9	4.0	5.7

The raw data collection is heterogeneous in the sense that it contains multiple document types. To characterize the heterogeneity, we randomly selected 2000 documents from the crawl repository, visually inspected them, and classified them into various categories [3]. The pie chart in Figure 1 indicates that over half of the crawled documents are academic, including papers, books, reports, slides, theses, abstracts, and posters, which can *potentially* be ingested. The rest (others+non-en) is about 42% including miscellaneous types such as advertisements. "Non-en" stands for documents not written in English or mixed with non-English languages (seen in some theses). The crawl repository can be used for document classification experiments and improving web crawling. URLs in the crawl database can be used to generate whitelists and schedule crawl jobs.

3.2 Production Databases

The processed data includes production databases and a repository. There are two principal databases, `citeseerx`

[1] http://csxcrawlweb01.ist.psu.edu

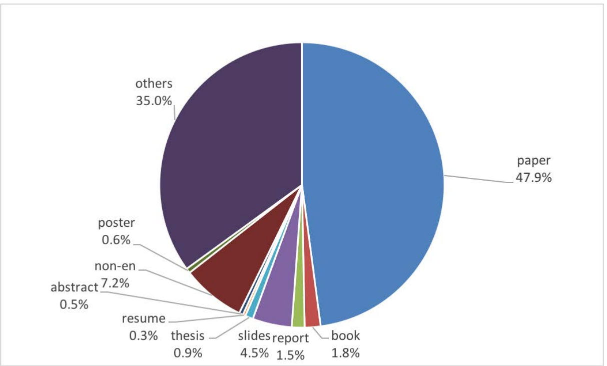

Figure 1: Categories for sampled crawled documents.

and `csx_citegraph`. The `citeseerx` database stores metadata directly extracted from papers. It also includes a table, `cannames`, for storing disambiguated authors. The `csx_citegraph` stores the citation graph – a citation relational network of publications after de-duplication [26]. The sizes of major tables are summarized in Table 2.

Table 2: Major database tables and their sizes by the end of 2015 in millions.

database.table	Description	Rows
citeseerx.papers	header metadata	6.8
citeseerx.authors	author metadata	20.6
citeseerx.cannames	authors (disambiguated)	1.2
citeseerx.citations	reference items in papers	150.2
citeseex.citationContext	citation context	131.9
csx_citegraph.clusters	citation graph (nodes)	45.7
csx_citegraph.citegraph	citation graph (edges)	112.5

Because each document cluster represents a unique publication, it is an eligible node in the citation graph. By the end of 2015, the citation graph contains about 45.7 million nodes, and 112.5 million edges. In Figure 2, we present the in-degree (k_{in}) and out-degree (k_{out}) distributions of this graph calculated using SNAP [14]. The slope by fitting data points of $0 \leq \log(k_{\text{in}}) \leq 2.5$ (data points beyond 2.5 introduce too much noise) using least square linear fit is $\gamma = -2.37$, significantly greater than the slope obtained by [2], which is -1.71, and generally consistent with the slope reported by [23], which is -2.28. Note that there are 0.7 million nodes and 1.7 million edges used by [23], and the work by [2] was based on a much smaller size of CiteSeer in 2004. The out-degree distribution is clearly non-linear, with a shallow slope ($\gamma \approx -0.22$) by fitting points with $k_{\text{out}} \leq 10$ and a steeper slope ($\gamma \approx -3.20$) beyond $k_{\text{out}} \geq 30$. Again, the high out-degree linear part is steeper than the slope reported by [2] ($\gamma = -2.32$ using points with $k_{\text{out}} > 18$) and more consistent with the slope reported by [23] ($\gamma = -3.82$), although they did not mention the fitting range.

The `citeseerx.cannames` table contains metadata of 1.2 million disambiguated authors, conflated from 20 million author names. Disambiguating all CiteSeerX authors takes less than 2 days [12]. It is non-trivial to directly evaluate the disambiguation results based on CiteSeerX data. Recently, we compared our algorithm with one of the best name disambiguation algorithms [25]. Our results outperformed all evaluation datasets in terms of recall and overall F-1 measure.

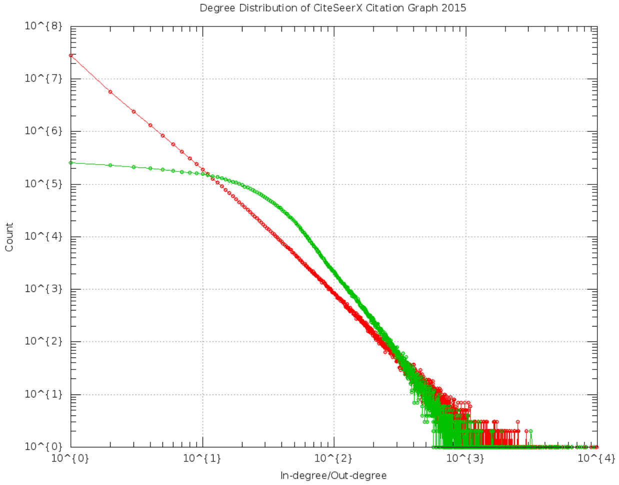

Figure 2: Distributions of in-degree (red) and out-degree (green) of the full citation citegraph.

3.3 Production Repository

Another data product is the production repository, containing about 7 million academic documents along with their metadata. The total size of the repository is about 9 TB. The PDF files takes 6.7 TB (5.4 TB compressed); the full text files takes 516 GB (165 GB compressed); the XML files takes 256 GB (31 GB compressed). The other files include body text (`.body`), reference text (`.cite`), original text extraction results from PDFLib TET (`.tetml`), version metadata files (e.g., `v1.xml`) and postscript files (`.ps` for old papers). The quality of the repository is evaluated by the accuracy of documents classified as academic and the rate of near-duplicates.

3.3.1 Classification accuracy

To evaluate the document classification accuracy, we randomly selected 1000 documents from the repository (Sample P) and manually classified them into categories defined in [3]. The results presented in Table 3 indicate that over 90% documents in CiteSeerX are academic and over 80% are papers. Non-academic documents are ingested because of the performance of our rule-based filter. As a result, a fraction of non-academic documents containing “references”, “bibliography”, and/or their variants are misclassified. Also, a fraction of academic documents is missing because the text does not contain designated terms. To quantify the fraction of false negatives, we randomly sampled 300 documents not identified as academic (Sample N) and manually classified them into the same categories as Sample P (Table 3). The results indicate that although the majority of dropped documents are non-academic (≈ 80%), the classifier misses a considerable fraction of academic documents (≈ 20%). To improve the classification accuracy, we have developed a more sophisticated binary and multi-type classifier using machine learning and structural features. The 10-fold cross validation results give over 90% of both precision and recall for binary classification tasks [3]. We have integrated this classifier into a new extraction framework [27].

3.3.2 Near-duplication rate

. The ingestion pipeline uses the keymapping algorithm

Table 3: Classification results of Samples P and N.

Categories	paper	book	report	slides	thesis
Sample P	83.0%	0.7%	4.5%	0.8%	2.6%
Sample N	12.3%	0%	0.7%	5.7%	0.3%

Categories	resume	abstract	poster	non-en	others
Sample P	0%	0.3%	0.2%	0.3%	7.5%
Sample N	0.7%	0.3%	0%	0.3%	70.7%

[29] to de-duplicate and cluster documents. Directly evaluating this approach and calculating the near-duplication rate is non-trivial due to the difficulty to build the ground truth by labeling documents in a sufficiently large and *unbiased* sample. Here, we infer and derive the near-duplication rate *indirectly* by labeling two samples. Sample A was drawn randomly from clusters containing exactly two documents ($S = 2$); Sample B was drawn randomly from clusters containing more than two documents ($S > 2$). We sampled in this way for two reasons. First, the number of $S = 2$ clusters is almost three times as many as the number of $S > 2$ clusters (Table 4). If we drew the sample from all $S \geq 2$ clusters uniformly, there would be too few $S > 2$ clusters, which leads to big uncertainties and extremely low confidence level. Second, the evaluation of $S = 2$ clusters is more straightforward than $S > 2$ clusters because the decision of the former is binary, but the latter involves cases in which not all documents are near-duplicates.

Table 4: Distribution of document cluster sizes.

S	1	2	3	4	> 4
$N_{\rm C}$ (million)	5.08	0.45	0.10	0.03	0.03
Percentage	92.8%	7.91%	1.76%	0.53%	0.53%

Each sample contains 100 clusters. Sample A contains 200 documents; sample B contains 430 documents. For each document, we manually extracted title, authors, year, and venue, if available. We visually inspected documents (not just metadata) in each cluster and judged if they are true near-duplicates. For Sample B, we calculate a partial-grade by dividing the number of correctly clustered documents by S. A cluster is then 100% correct if and only if all documents are true near-duplicates. Table 5 shows that the larger the cluster size, the more likely the documents inside are *not* near-duplicates, i.e., they are distinct documents. The table column "D-ratio" is defined as the number of distinct documents D divided by the number of clusters $N_{\rm C}$. D-ratio values in Samples A and B mean that assuming we correctly de-duplicate all documents, we should gain 16% more clusters in Sample A, and 126% more clusters in Sample B. The number of distinct documents of the whole repository can then be estimated by scaling up by D-ratios. Assuming all the $S = 1$ clusters are distinct, the total number of unique documents is estimated as (in millions) $5.08+0.45\times1.16+0.16\times2.26 \approx 5.96$, so the duplication rate is $(1-5.96/6.70)\times100\% \approx 11\%$. Here we assume there is no cross-cluster near-duplicates, i.e., there is no duplicate of any document in Cluster X with Cluster Y.

Here we argue that the clustering quality does not only depend on the algorithm, but also on extraction quality. Our new extraction framework employs GROBID [18], which exhibits superior performance over SVMHeaderParse [17] on header extraction. We expect the keymapping algorithm achieve better performance with improved metadata.

Table 5: Near-duplicate evaluation of two samples.

Sample	S	$N_{\rm C}$	True	%True	D-ratio
A	2	100	84	84%	1.16
B overall	> 2	100	70	70%	2.26
B	3	58	44	76%	1.40
B	4	27	22	81%	1.44
B	5	5	2	40%	2.00
B	6	3	0	0%	4.67
B	7	1	0	0%	5.00
B	8	2	1	50%	4.50
B	9	1	0	0%	8.00
B	13	1	1	100%	1.00
B	21	1	0	0%	20.00
B	39	1	0	0%	39.00

3.4 Data Management and Access

Scalability is the top concern of scholarly big data. To keep our data highly available through the Web service, we use a dedicated server with 48GB of RAM, 16 cores, and 1TB storage to host the master database. The database is replicated in real time on another two servers. The search data is hosted by Apache Solr 4.9 hosted on two high end servers, one replicating the other. We have also deployed *SolrCloud* using 7 virtual servers on a private cloud, which will be in production after tuning. SolrCloud is extremely scalable up to hundreds of millions of documents, and has been leveraged by industrial companies. The repository used to be handled as a bulk device in GFS2. Recently, we designed and implemented a RESTful API, which overcomes the stability issue caused by the fencing function of GFS2.

Data redundancy and backups are crucial for an information system like us as re-producing the data takes incredibly long period of time and some data are not restorable once they are lost. The database is dumped periodically using the standard MySQL tool in `.sql` format, which keeps data integrity and compatibility. The production repository is synced to a backup repository on a weekly basis. We keep at least three copies of production repository and database dump, and two copies of the crawl repository, as well as other related data. We will keep applying these strategies to future data. The production data is accessible from Amazon S3, which is updated every 2–3 months. All data requesters provide basic information such as their names, institutions, and a brief explanation of intended use by submitting a "contact us" message on the CiteSeerX front page.

4. SEMANTICIZATION

DLSEs are evolving to be intelligent to question answering systems with regards to concepts, experts, methodologies, and paper and citation recommendations. An appropriate question could be "What papers should I read to understand digital library search engines?". Large scale commercial search engines such as Google have built semantic search engines capable of answering a number of general questions. There has been work on semanticizing syntactic patterns in NLP processing [24], but it is usually based on tagged sentences, which are preprocessed by lexical or semantic parsers. Such a semantic search engine is a knowledge base populated with entities extracted from full text or from other metadata. For our data, we utilize the UIUC Wikifier [5, 19] for entity linking on a sample of CiteSeerX papers. There are similar tools such as *Semanticizer* [8]. but we choose the UIUC Wikifier because of its relatively stable performance and wide usage. As an open source tool, Wikifier has been used and studied in various entity extraction tasks [16, 20, 30]. Basically it identifies entities and disambiguates them into the most corresponding Wikipedia pages based on local and global statistics of the given text and entity relations.

We ran Wikifier on 24859 paper full text randomly selected from CiteSeerX repository, of which 21300 are successfully processed. The average size of an input file is 53 kB. The output of each file includes all Wikipedia terms identified, along with a link score. We set an empirical cut-off of 0.8 to remove less meaningful terms, e.g., "Symbols_(album)", and removed single character symbols such as "Ω" since we are focusing on words. Figure 3 shows the frequency-rank (f–R) diagram of about 280,000 Wikipedia entities extracted. The entities do not follow Zipf's law. Instead, the term frequency drops quickly at $R > 1000$. The top 10 frequent terms are "Algorithm", "Cell_(biology)", "Matrix_(mathematics)", "Protein", "United_States", "Energy", "Temperature", "One_half", "Need_To", and "Theorem".

General knowledge extraction focuses on high frequency entities. Facts extracted based on these entities are more general and less context dependent. However, for scholarly papers, such a fact extractor may not work well because statements are more meaningful when they are placed in a global or local context. This is partially (if not all) attributed to the fact that many domain specific terms are not in Wikipedia, e.g., "Digital Library Search Engine". The hard drop at $R > 1000$ indicates that only considering terms appearing in Wikipedia is not sufficient to cover the full spectrum of meaningful entities. The missing entities are likely to be very contextual and domain dependent. Thus, it is necessary to curate a set of domain specific terms, which will appear at low frequency that are crucial for making meaningful extractions.

The poor scalability of the extractor limited experiments on a larger sample. We ran our experiment on a Linux server with 24 cores and 48 GB of RAM. Only a small fraction of runtime was parallelizable. To run the extractor on all CiteSeerX data may require optimization or launching multiple instances. Another issue is that the tools are trained with annotated news datasets or the Wikipedia dataset. A labeled entity linking dataset for scholarly papers would be useful.

To build a semantic entity extractor for scholarly papers, we start from a sample of about 30,000 conference and journal papers published in WWW, VLDB, and ACL. High quality text is first extracted with Xpdf [7]. The full text of each paper is parsed to n-grams with $n = 1 \cdots 5$. A 10-dimensional feature vector is then created for each n-gram, including *tf*, *df*, *tf-idf*, *first letter capitalized*, *all letters capitalized*, *appearance of citation*, *mixed upper/lower cases*, *generalized dice coefficient (GDC)*, *gram locations*, and *Pointwise Mutual Information (PMI)*. By filtering out n-grams containing non-alphanumerical characters, we have a list of 8000 candidate entities. The next step is to label them to build a ground-truth dataset so as to enable model training.

5. CONCLUSION AND FUTURE WORK

We describe a corpus of scholarly big data released by CiteSeerX, and report on preliminary results for extracting semantic entities from a sample of about 25,000 papers. Future work will be to significantly increase the size of the data to 7 million and hopefully to all open access documents on the Web (about 30 million). The data variety will be improved by extracting non-textual content such as figures, tables, and algorithms using many developed tools, including ours. We will improve data quality using better extractors and investigate empirically what are "meaningful" terms. We will develop a data correction module, which automatically detects and corrects metadata errors using available reference data available from DBLP, publishers, and the Web of Science.

We believe better semanticization can come from an annotated a corpus of scholarly papers used as training data. We intend to investigate the difference and similarity of papers of various subjects, such as computer science, physics, and chemistry in order to find a proper set of semantic entities for a specific knowledge domains before establishing their semantic relations. Also, we intend to construct a fine grained hierarchical ontology for computer science papers, for which many of our semantic entities are based.

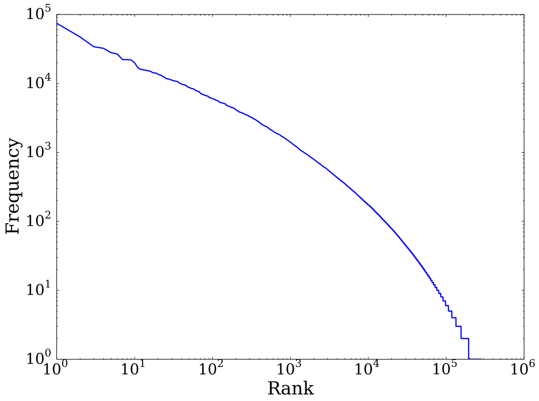

Figure 3: The frequency-rank diagram of about 280,000 Wikipedia entities extracted from a text sample selected from CiteSeerX repository.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge partial support from the National Science Foundation and AllenAI.

7. REFERENCES

[1] B. Aleman-Meza, F. Hakimpour, I. B. Arpinar, and A. P. Sheth. Swetodblp ontology of computer science publications. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(3):151 – 155, 2007.

[2] Y. An, J. Janssen, and E. E. Milios. Characterizing and mining the citation graph of the computer science literature. *Knowledge and Information Systems*, 6(6):664–678, 2004.

[3] C. Caragea, J. Wu, S. D. Gollapalli, and C. L. Giles. Document Type Classification in Online Digital Libraries. Phoenix, Arizona USA, 2016. AAAI.

[4] H.-H. Chen, P. Treeratpituk, P. Mitra, and C. L. Giles. CSSeer: an expert recommendation system based on CiteSeerX. JCDL '14, pages 381–382, 2013.

[5] X. Cheng and D. Roth. Relational inference for wikification. In *EMNLP*, 2013.

[6] I. Councill, C. L. Giles, and M.-Y. Kan. Parscit: an open-source crf reference string parsing package. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.

[7] FooLabs. http://www.foolabs.com/xpdf/index.html. Accessed 06-May-2016.

[8] D. Graus, D. Odijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Semanticizing search engine queries: The university of amsterdam at the erd 2014 challenge. In *Proceedings of the First International Workshop on Entity Recognition &#38; Disambiguation*, ERD '14, pages 69–74, New York, NY, USA, 2014. ACM.

[9] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines. In *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '03, pages 37–48, 2003.

[10] W. Huang, Z. Wu, P. Mitra, and C. L. Giles. Refseer: A citation recommendation system. In *IEEE/ACM Joint Conference on Digital Libraries, JCDL 2014, London, United Kingdom, September 8-12, 2014*, pages 371–374, 2014.

[11] M. Khabsa and C. L. Giles. The number of scholarly documents on the public web. *PLoS ONE*, 9(5):e93949, May 2014.

[12] M. Khabsa, P. Treeratpituk, and C. Giles. Large scale author name disambiguation in digital libraries. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 41–42, Oct. 2014.

[13] P. Larsen and M. von Ins. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, 84(3):575–603, 2010.

[14] J. Leskovec and R. Sosič. Snap.py: SNAP for Python, a general purpose network analysis and graph mining tool in Python. http://snap.stanford.edu/snappy, June 2014.

[15] M. Ley. DBLP - some lessons learned. *PVLDB*, 2(2):1493–1500, 2009.

[16] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1070–1078. ACM, 2013.

[17] 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.

[18] 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.

[19] L.-A. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL*, 2011.

[20] A. Sil and A. Yates. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on information & knowledge management*, pages 2369–2374. ACM, 2013.

[21] R. Sinatra, P. Deville, M. Szell, D. Wang, and A.-L. Barabasi. A century of physics. *Nat Phys*, 11(10):791–796, 10 2015.

[22] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. P. Hsu, and K. Wang. An Overview of Microsoft Academic Service (MAS) and Applications. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 243–246, Republic and Canton of Geneva, Switzerland, 2015.

[23] L. Subelj, D. Fiala, and M. Bajec. Network-based statistical comparison of citation topology of bibliographic databases. *Scientific Reports*, 4:6496, Sep 2014. Article.

[24] N. Vitucci, M. A. Neri, R. Tedesco, and G. Gini. Semanticizing syntactic patterns in NLP processing using SPARQL-DL queries. *CEUR Workshop Proceedings*, 849, 2012.

[25] M. Wick, S. Singh, and A. McCallum. A Discriminative Hierarchical Model for Fast Coreference at Large Scale. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 379–388, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[26] K. Williams and C. L. Giles. Near duplicate detection in an academic digital library. DocEng '13, pages 91–94, 2013.

[27] J. Wu, J. Killian, H. Yang, K. Williams, S. R. Choudhury, S. Tuarob, C. Caragea, and C. L. Giles. Pdfmef: A multi-entity knowledge extraction framework for scholarly documents and semantic search. In *Proceedings of the 8th International Conference on Knowledge Capture*, K-CAP 2015, pages 13:1–13:8, New York, NY, USA, 2015. ACM.

[28] 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.

[29] 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.

[30] J. G. Zheng, D. Howsmon, B. Zhang, J. Hahn, D. McGuinness, J. Hendler, and H. Ji. Entity linking for biomedical literature. *BMC medical informatics and decision making*, 15(Suppl 1):S4, 2015.