Utility-Based Control Feedback in a Digital Library Search Engine: Cases in CiteSeerX

Jian Wu[†] *Alexander Ororbia*[†] *Kyle Williams*[†] *Madian Khabsa*[‡] *Zhaohui Wu*[‡]

C. Lee Giles[†‡]

[†]*Information Sciences and Technology*

[‡]*Computer Science and Engineering*

Pennsylvania State University, PA, 16802 USA

Abstract

We describe a utility-based feedback control model and its applications within an open access digital library search engine – CiteSeerX, the new version of CiteSeer. CiteSeerX leverages user-based feedback to correct metadata and reformulate the citation graph. New documents are automatically crawled using a focused crawler for indexing. Those documents that are ingested have their document URLs automatically inspected so as to provide feedback to a whitelist filter, which automatically selects high quality crawl seed URLs. The changing citation count plus the download history of papers is an indicator of ill-conditioned metadata that needs correction. We believe that these feedback mechanisms effectively improve the overall metadata quality and save computational resources. Although these mechanisms are used in the context of CiteSeerX, we believe they can be readily transferred to other similar systems.

1 Introduction

While classic feedback control has been widely used in mechanical systems [14], integrated circuits [7], etc., there is an emerging interest in applying feedback control to computing systems, e.g., [16], [9] and [18].

The fundamental tenets of feedback computing are from control theory and self-regulating, adaptive systems [21]. Given such an information-centered view, it follows that self-correcting systems, often complex and modular in nature, change their internal state in response to a stimulus from the environment. Although these concepts have largely been used to describe biological entities or social constructs, a complex, adaptive system could very well be an entity such as a digital library, built with the primary aim of automatically collecting and organizing vast arrays of unstructured content retrieved from the World Wide Web. In this context, the notions of adaptation and information can be related to dynamic resource management, where automated techniques are employed to alter the system state configuration in response to fluctuations in workload and error cases [15]. We reinterpret feedback computing as user-based feedback which is useful in improving a digital library search engine.

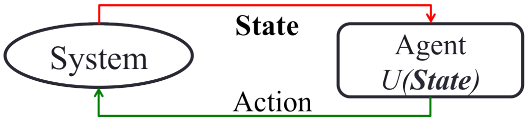

Figure 1: The utility-based control feedback loop.

To represent high-level policies, a utility function $U(\mathbf{S})$ is defined [17], which maps any possible system state, expressed in terms of a service attribute vector **S**, to a scalar value [22]. An agent can be regarded as a controller, which adapts by learning the mapping from actions to service level attributes, applying the utility function, and choosing the action that maximizes utility. This type of feedback loop is illustrated in Figure 1. More importantly, this type of *utility-driven* management architecture has been applied to the design of CiteSeerX a scholarly digital library search engine [12, 25].

For clarification, we give a brief explanation of how CiteSeerX works. A web crawler is used to harvest scholarly documents in PDF from the web and other public resources. These documents are then converted to ascii and checked to see if they are indeed scholarly documents and for duplicates. Various metadata is automatically extracted and the text is indexed with a focus on automatic citation indexing. A similar such system is Google Scholar. Here, we introduce three applications for different paradigms that fall under this utility-driven model. How these applications fit into the utility-driven model is summarized below.

User-correction CiteSeerX allows registered users to directly correct paper metadata from the web interface. In this case, the "system" is the CiteSeerX web service and database; the "agent" is the group of registered CiteSeerX users; the "state" is a vector comprised of paper metadata fields, and the "action" is the user correction behavior. This is primarily based on user-based feedback (Section 3.1).

Ill-conditioned Metadata Detection By checking the citation and download history, we can detect papers whose metadata are ill-conditioned, i.e., some critical fields contain wrong values. In this case, the "system" is the CiteSeerX database; the "agent" is the metadata checking program; the "state" is a two-element vector containing the citation and downloading counts, and the "action" is an automated metadata correction program. This is based on long-term feedback (Section 3.2).

Crawl Whitelist Generation CiteSeerX automatically downloads PDF documents using a focused web crawler. The seeds injected to the crawler come from a whitelist containing high quality URLs selected on the number of ingestable documents, which are documents classified as scholarly papers. In this case, the "system" is the focused crawler and document classifier; the "agent" is the whitelist generator; the "state" is the number of ingestable documents of a parent URL, and "action" is the whitelist URL selection process. This is based on automated feedback from the document filter (Section 2).

Because feedback plays an important role in digital library search engines, we plan to extend current feedback control implementations. For example, we are developing a crawl scheduler which makes use of feedback from the document filter *and* crawl history to increase the freshness of CiteSeerX (Section 5) documents.

2 Related Work

Given more practical, engineering-oriented objectives, such as workload balancing, feedback systems such as the Alexandria Digital Library have been modified to essentially work like feedback computing systems [26]. In particular, this digital library was extended to leverage novel optimization techniques that would allow the system to dynamically self-monitor resource availability among its CPU cluster nodes and allow scheduling requests based on a predictive sampling of its internal state. Furthermore, a complex computational engine such as a digital library would need to be robust and responsive to better serve its user base and make itself easier to use [6] plus learn from its mistakes or episodes of suboptimal service to improve future performance.

The actual harvesting of unstructured scholarly documents from the web, which constitutes the primary information source of a CiteSeerX-like digital library engine, can also be viewed from the perspective of feedback computing, in particular focused crawling [19]. In [3], feedback came from user inspection of the crawler's harvest, where the agent would report the "most popular sites and resource lists" and the user would mark these items as relevant or not, which would then be used by the agent to improve topical crawling performance. Ant-based crawling [10] and hybrid evoluationary-reinforcement learning schemes [20] for adaptive topic-driven crawling also take this notion of feedback to the extreme. Here, crawling agents reproduce and die based on the positive or negative stimulus received when evaluating how well they satisfy the content relevance objective function.

In essence, the variety of ways in which complex systems like digital libraries can harness information to self-modify their behavior allows them to remain relevant in rapidly changing, uncertain environments. CiteSeerX-exhibits many of these feedback-driven characteristics and is an example of flexible technology that scales and adapts to the web.

3 Metadata Correction

Metadata plays important roles in digital libraries and is used for organizing data, enabling specialty search, and record linking. Within scholarly digital libraries, metadata usually refers to information such as titles, authors, years, venues, and venues of papers. Some digital libraries, such as the arXiv, require users to enter metadata as part of the submission process. CiteSeerX, on the other hand, automatically extracts metadata in XML from scholarly documents. This has the benefit that it is much more scalable than manual information extraction. However, this comes at the cost of accuracy and metadata quality.

In CiteSeerX there are two standard types of metadata: header and citations[23]. Header metadata refers to the information that is contained within the header of a document, such as the title, authors, affiliations, abstract, and journal or conference. This information is currently extracted using an SVM-based header extraction tool [13] and citations are extracted using a CRF-based parsing and tagging tool [8]. Other metadata such as acknowledgement, pseudo-code, figures and tables can also be extracted [1]

Near duplicate (ND) documents are common in scholarly digital libraries and, as a result, their header metadata are used to group ND documents into clusters. A document cluster is a representation of a paper entity regardless of its versions. Because of this, the citation graph uses clusters as nodes rather than individual papers. Clusters are useful in that they allow accurate calculation of statistics. For example, citation counts must be calculated at the cluster level rather than the individual document level and download statistics for a cluster are better reflections of the popularity of a paper.

3.1 User Correction

The CiteSeerX web interface offers a feature to allow registered users to correct metadata mistakes online. These metadata mistakes often appear during the text and metadata extraction. Users can manually change almost all metadata fields and the new values are effective after the changes are submitted. The metadata of a document can be changed multiple times and CiteSeerX generates a new XML metadata file for each metadata version and saves it into the document repository.

After a user correction (uC) is submitted, the paper's initial cluster is deleted. Papers in this cluster are reclustered and the citation graph is reformulated based on the revised metadata. Feedback from uC allow CiteSeerX to obtain for free high quality metadata of important papers, which can be further used to improve the quality of the automated metadata extraction.

Since the uC feature was activated, CiteSeerX has received more than 277,000 user corrections on approximately 251,000 papers. To get a preliminary evaluation of the user feedback, we randomly select 50 uC instances and compared the metadata values before and after corrections. Each metadata field that is changed is tagged with one of four correction types: WC (real corrections from wrong to correct), WW (wrong to wrong; metadata quality not improved), CW (correct to wrong; metadata quality decreases) and WD (a wrong value is deleted; no metadata provided). Note that not all users make correct changes. Some simply delete the content of a wrong field. We do not see any uCs in our sample where users delete a correct field.

The tagging results are tabulated in Table 1. In the last row, we count the number of uCs (in our sample, it is equivalent to the number of papers since all papers are unique) that are tagged with a specific correction type. For example, the first number in the last row means that the metadata fields are changed from wrong to correct in 45 uCs. In the rows above the last row, we break down the total count by examining the correction type for each field. For example, the first number in the "title" row means that the title field was changed from wrong to correct for 23 uCs.

Table 1: Tagging results of uC samples.

metadata field	WC	WW	CW	WD
title	23	1	0	0
abstract	21	1	2	4
name	19	0	1	0
year	15	0	0	1
affiliation	13	0	0	1
address	7	0	0	1
email	7	0	0	0
venue	5	0	2	0
pages	4	0	0	0
publisher	2	0	1	0
venType	2	0	2	1
pubAddress	1	0	0	0
volume	1	0	1	1
tech	1	0	0	1
author	0	0	0	8
#uCs tagged	45	2	5	13

tech: technical report description; *author*: an author block, including author name, affiliation, address and email.

The uC evaluation results show, as seen in the last row, that most uCs (90%) contain real corrections. While there are a few uCs that do not improve the metadata quality, 13 uCs involve metadata deletions, most of which are author blocks and abstracts. Breaking the uC count for each tag, we see that titles, abstracts, author names and years are the most corrected fields, which is expected since they are the important fields of a paper. There are also many changes to author affiliations (13 out of 50). Changes to the remaining fields are not so common. Some additional author blocks are deleted. A small number of abstracts are deleted. It is also noted that most papers are corrected only one time[1], so the number of corrected papers is roughly equal to the number of uCs. If we scale the fractions of the random sample to all uC records, about 116,000 paper titles are corrected, which is about 3% of all CiteSeerX papers. Author names are corrected in about 100,000 papers, which is about 2.5% of all CiteSeerX papers. Although the fractions are small, these are among the most downloaded papers.

3.2 Ill-conditioned Metadata Detection

User access logs have been used to provide implicit feedback for many aspects of search engines, especially for learning relevancy and tweaking ranking functions. In a digital library and search engine such as CiteSeerX where metadata is automatically extracted, errors are inevitably introduced. As such, access logs can be used to detect anomalies and errors in metadata extraction.

[1] Corrections performed by system maintenance do not count.

Log analysis showed a positive correlation between the number of times a given paper is downloaded and the number of citations it has received. Given this correlation, an anomaly in log behavior showed that certain papers that had download requests that placed them in the top 1% of the most downloaded papers had zero citations in the citation graph database. Manual inspection of these papers found that many had their metadata incorrectly extracted by the header extractor. This resulted in mis-assigned citations to these papers because the citation matching algorithm relies on title information. Therefore, when a paper is found to receive a large number of downloads over an extended period of time without being cited at least once, it is marked as potentially having erroneous metadata. These marked documents are then assessed using different sources of metadata, such as a publisher website or a secondary digital library, for possible corrections.

4 Crawl Whitelist Generation

CiteSeerX uses a focused web crawler to automatic harvest PDF documents from the Web. Different from crawlers used by general search engines in which nearly everything is downloaded, the CiteSeerX crawler targets scholarly documents in PDF formats. [24] showed that seed quality is essential for the efficiency of a focused crawler. As such CiteSeerX uses a whitelist crawling policy, which has two parts:

1. A whitelist is crawled containing high quality URLs selected from previous crawled URLs.
2. URLs outside of the domains defined by the whitelist are not crawled.

While the second can be achieved by configuring the crawler (e.g., Heritrix), the whitelist is generated offline by evaluating each parent URL crawled beforehand. A parent URL is a hyperlink of a webpage, which contains the document URLs directly linking to the actual PDF files. CiteSeerX stores all previously crawled parent URLs, each of which contains at least one document URL. If we generate the whitelist by counting only the number of PDF document URLs (n_{d}), we may download a large fraction of non-scholarly documents. This is not only a waste of bandwidth but also of disk space and computational resources until they are filtered out by the document classifier. Therefore, n_{d} is not a good form of feedback.

Since we are particularly interested in scholarly documents, it is intuitive to use the number of ingested documents, n_{ind}, as the feedback stimulus to generate the whitelist. Ingestion is the process of writing the extracted metadata of scholarly documents into the production database, so that users can view and correct them (if necessary). CiteSeerX uses a rule-based filter to determine whether a document is or is not scholarly by searching keywords/keyphrases over the entire text body. For each crawled document, the filtering result is represented by a binary flag in the crawl database. A parent URL is selected and placed into the whitelist if its corresponding n_{ind} is equal or greater than one.

For evaluation, we compare the crawl efficiencies before and after incorporating the feedback. We run two sets of experiments, Set P and Set W. In each experiment of Set P, we crawl 500 seed URLs randomly selected from 200,000 URLs in the parent URL table. In each experiment of Set W, we crawl 500 seeds randomly selected from the whitelist containing 46782 URLs generated out of the 200,000 parent URLs above. To reduce errors introduced by small sample sizes, we run 10 crawl experiments in each set before performing text extraction and filtering.

The results are tabulated in Table 2. Although the absolute number of n_{ind} in Set P is higher in general, the average fraction of ingestable documents in Set W ($44.83 \pm 12.14\%$) is significantly higher than that of Set P ($22.87 \pm 5.04\%$)[2]. Given that the crawling time is correlated with n_{d}[3], the larger fraction in Set W indicates that it can crawl more ingestable documents within a given period of time. This feedback increases the crawl efficiency by at least 20%.

Table 2: Crawl efficiency comparison.

#	$n_{\mathrm{ind}}/n_{\mathrm{d}}$ (Set P)	$n_{\mathrm{ind}}/n_{\mathrm{d}}$ (Set W)
1	6905/29276(23.58%)	698/1308(53.36%)
2	2088/8924(12.90%)	1152/1735(66.38%)
3	3784/16186(23.38%)	575/1668(34.47%)
4	2438/11141(21.88%)	1002/2413(41.52%)
5	2740/13974(19.61%)	2362/3951(59.78%)
6	2259/9395(24.04%)	2126/4850(43.84%)
7	1845/9873(18.69%)	1498/3298(45.42%)
8	3089/9432(32.75%)	1252/4606(27.18%)
9	2079/7486(27.77%)	1214/4316(28.13%)
10	1998/8284(24.12%)	1298/2694(48.18%)

N_{d}: the number of PDF documents crawled.
N_{ind}: the number of scholarly documents (ingestable documents).

This control mechanism is stable not only because the seed URLs are selected from the whitelist but also that

[2] Error bars are calculated assuming the experimental results are a Gaussian distribution.

[3] Although there is a significant number of HTML files, the bandwidth is still dominated by PDF files.

the subsequential URLs are restricted to be within the whitelist domain. However, this may cause coverage issues since the crawler does not explore domains outside. If we remove the domain constraints, can the control still provide a stable output? In this situation, when the crawler sees a URL that is neither in the blacklist nor in the whitelist domain, it crawls and writes it to the crawl database if it contains any PDF documents. Given the fact that we have no information on the quality of these URLs, the crawl efficiency may vary and drop in the long term because non-scholarly documents will most likely dominate the crawled PDFs. We thus expect the mean efficiency $n_{\text{ind}}/n_{\text{d}}$ should drop below 44.83% with an increase in domain coverage. Because of the feedback from the document filter, high quality URLs are selected as starting points, the focused crawl efficiency would still be higher than 22.87%.

5 Future Work

5.1 Crawl Scheduler

One extension of the crawling module in CiteSeerX would be to integrate a crawl scheduler that utilizes feedback from both the filtering results and the crawl history so that the URL state $\mathbf{S} = \mathbf{S}(n_{\text{ind}}, \lambda)$, in which λ is the mean updating rate of the parent URL web page. The value of λ affects the re-crawl frequency while the filtering results determine the URLs in a whitelist.

To estimate λ based on crawl history, we assume the change of a parent URL webpage is a Poisson process. Experiments reported in previous studies indicated that the changes to many webpages follow a Poisson process[4], e.g., [2, 4]. The key to estimating the updating rate λ is to find a good estimator from repeated accesses to a parent URL. An estimator proposed by [5] is

$$\hat{\lambda} = -\log\left(\frac{\bar{X}+0.5}{n+0.5}\right) \tag{1}$$

in which $\hat{\lambda}$ is an estimation of the updating frequency, $\bar{X} = n - X$ is the number of accesses in which the webpage did *not* change, X is the number of detected changes, and n is the number of accesses within a time period of T. One thing we note is that the larger the n is, the less biased the estimator. From the crawling history, we can identify if n_{ind} and/or the actual PDF files are changed for a given parent URL. We can then calculate $\hat{\lambda}$. The next time this parent URL is re-crawled is $I \approx \hat{\lambda}$. Based on this estimator, the whitelist URLs can be generated on a daily basis. This scheduler can significantly increase the freshness of our document collection. Note that I may dynamically change over time.

An example of this crawling model [11] actively learns a changing page-update distribution from the (page history) data at a given moment (usually after various ”cycles”) and makes decisions as to which pages it should revisit in order to maintain content freshness. The more page-history data the agent collects, the more reliable and refined it becomes.

5.2 Dynamical Topic-Driven Crawling

Another useful modification of the CiteSeerX crawler would be to design and incorporate adaptive crawling agent models. While more artificial intelligent approaches to focused crawling are under investigation, most of these models are static after an initial learning/training phase. However, this means that these agents' ability to detect relevant scholarly publications quickly deteriorates as the web changes over time.

A fruitful direction could be to explore the viability of multi-agent models, such as the ant-colony focused crawling algorithm mentioned in Section 2. The motivation for using feedback-centric agent models stems from the fact that new web data, mostly unlabeled, is being generated at an increasing rate. The manual labeling of useful and representative example documents (which statistical learning algorithms can be trained on) doesn't scale. Crawling agents, perhaps endowed with an initial inductive bias, that can interact with each other and make use of unlabeled data to learn from previous mistakes, pass on this knowledge to other agents, and modify their internal knowledge to gather yet more relevant content from the web should be very effective. Further, this adaptive behavior would not only lead to more accurate, long-term relevance mining but also reduce computational and network resource usage.

6 Summary

We described three applications of the utility-based control feedback model for scholarly digital library search engine. The user-correction allows registered users to perform online changes to metadata. In more than 90% of cases, the users provides correct changes, which improves the metadata quality of the many highly downloaded papers. The downloading and citation history can be used as feedback to detect ill-conditioned metadata, which helps make automatic corrections and improve metadata extraction. The whitelist generator utilizes positive and negative feedback from the document filter to decide which URLs to use. This feedback mechanism boosts the crawl efficiency (the fraction of ingestable documents) by at least 20%. All three applications indicate that feedback controls are useful for a

[4] Not all webpages follow the Poisson process

CiteSeerX-like digital libary search engine, whose primary data come from crawling the Web. Future work would be to quantify the significance and importance of the existing feedback. A more advanced application would be to design a multi-agent focused crawler that self-monitors its success in finding scholarly papers and adaptively changes crawling behavior in real-time.

7 Acknowledgments

This work has been partially supported by the National Science Foundation.

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 citeseer^X digital library. *D-Lib Magazine 18*, 7/8 (2012).

[2] Brewington, B. E., and Cybenko, G. How dynamic is the web? In *Proceedings of the 9th International World Wide Web Conference on Computer Networks : The International Journal of Computer and Telecommunications Netowrking* (Amsterdam, The Netherlands, The Netherlands, 2000), North-Holland Publishing Co., pp. 257–276.

[3] Chakrabarti, S., Punera, K., and Subramanyam, M. Accelerated focused crawling through online relevance feedback. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, ACM, p. 148159.

[4] Cho, J., and Garcia-Molina, H. The evolution of the web and implications for an incremental crawler. In *Proceedings of the 26th International Conference on Very Large Data Bases* (San Francisco, CA, USA, 2000), VLDB '00, Morgan Kaufmann Publishers Inc., pp. 200–209.

[5] Cho, J., and Garcia-Molina, H. Estimating frequency of change. *ACM Trans. Internet Technol. 3*, 3 (Aug. 2003), 256–290.

[6] Chowdhury, G. G., and Chowdhury, S. Digital library research: major issues and trends. 409–448.

[7] Choy, C., Chan, C., and Ku, M. H. A feedback control circuit design technique to suppress power noise in high speed output driver. In *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on* (Apr 1995), vol. 1, pp. 307–310 vol.1.

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

[9] Diao, Y., Gandhi, N., Hellerstein, J., Parekh, S., and Tilbury, D. Using mimo feedback control to enforce policies for interrelated metrics with application to the apache web server. In *Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP* (2002), pp. 219–234.

[10] Dzitac, I., Moisil, I., Mastorakis, N. E., Poulos, M., Mladenov, V., Bojkovic, Z., Simian, D., Kartalopoulos, S., Varonides, A., and Udriste, C. Advanced AI techniques for web mining. In *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, WSEAS.

[11] Edwards, J., McCurley, K., and Tomlin, J. An adaptive model for optimizing performance of an incremental web crawler. In *Proceedings of the 10th international conference on World Wide Web*, ACM, p. 106113.

[12] Giles, C. L., Bollacker, K. D., and Lawrence, S. Citeseer: An automatic citation indexing system. In *ACM DL* (1998), pp. 89–98.

[13] Han, H., Giles, C., Manavoglu, E., Zha, H., Zhang, Z., and Fox, E. Automatic document metadata extraction using support vector machines. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries* (2003), pp. 37–48.

[14] Hehn, M., and D'Andrea, R. A flying inverted pendulum. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (May 2011), pp. 763–770.

[15] Hellerstein, J. L., Diao, Y., Parekh, S., and Tilbury, D. M. *Feedback Control of Computing Systems*. John Wiley & Sons.

[16] Hellerstein, J. L., Gandhi, N., and Parekh, S. S. Managing the performance of lotus notes: A control theoretic approach. In *Int. CMG Conference* (2001), pp. 397–408.

[17] Kephart, J., and Walsh, W. An artificial intelligence perspective on autonomic computing policies. In *Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on* (June 2004), pp. 3–12.

[18] Lemmon, M. Towards a passivity framework for power control and response time management in cloud computing. In *Proceedings of 7th International Workshop on Feedback Computing* (2012).

[19] Micarelli, A., and Gasparetti, F. Adaptive focused crawling. In *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., no. 4321 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 231–262.

[20] Pant, G., and Menczer, F. MySpiders: evolve your own intelligent web crawlers. 221–229.

[21] Von Bertalanffy, L. General system theory: foundations, development, applications (revised edition).

[22] Walsh, W., Tesauro, G., Kephart, J., and Das, R. Utility functions in autonomic systems. In *Autonomic Computing, 2004. Proceedings. International Conference on* (May 2004), pp. 70–77.

[23] Williams, K., Wu, J., Choudhury, S. R., Khabsa, M., and Giles, C. L. Scholarly Big Data Information Extraction and Integration in the CiteSeerX Digital Library. IIWeb '14.

[24] Wu, J., Teregowda, P., Ramírez, J. P. F., Mitra, P., Zheng, S., and Giles, C. L. 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* (New York, NY, USA, 2012), WebSci '12, ACM, pp. 340–343.

[25] 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.

[26] Zhu, H., Yang, T., Zheng, Q., Watson, D., Ibarra, O. H., and Smith, T. Adaptive load sharing for clustered digital library servers. *International Journal on Digital Libraries 2*, 4 (2000), 225–235.