Indexing and Retrieval of Scientific Literature

Steve Lawrence, Kurt Bollacker, C. Lee Giles
NEC Research Institute, 4 Independence Way, Princeton NJ 08540
{lawrence,kurt,giles}@research.nj.nec.com

Abstract

The web has greatly improved access to scientific literature. However, scientific articles on the web are largely disorganized, with research articles being spread across archive sites, institution sites, journal sites, and researcher homepages. No index covers all of the available literature, and the major web search engines typically do not index the content of Postscript/PDF documents at all. This paper discusses the creation of digital libraries of scientific literature on the web, including the efficient location of articles, full-text indexing of the articles, autonomous citation indexing, information extraction, display of query-sensitive summaries and citation context, hubs and authorities computation, similar document detection, user profiling, distributed error correction, graph analysis, and detection of overlapping documents. The software for the system is available at no cost for non-commercial use.

1 Introduction

The progress of science has often been hampered by the inefficiency of traditional methods of disseminating scientific information. Publication delays, and the difficulty in easily locating all relevant literature, mean that researchers are not always working with the most up-to-date and comprehensive information available. The World Wide Web, along with search engines such as AltaVista, have greatly improved the dissemination and retrieval of an increasingly large body of information [1, 20, 21]. However the major web search engines such as AltaVista do not index the content of a large body of scientific literature on the publicly indexable web: Postscript or PDF copies of research articles. This article discusses the creation of an index of scientific literature on the web, called *CiteSeer*, along with several features that improve access to scientific literature.

The purpose of this paper is to outline the CiteSeer project, to provide details of several aspects of the project not contained in the previous papers that focus on the citation indexing component [3, 13], and to encourage work on the CiteSeer project or related projects (the software and data from CiteSeer is available at no cost for non-commercial use).

2 Related Work

There are many freely available indices of scientific literature on the web, examples include the LANL e-Print archive, NCSTRL, UCSTRI, ML Papers, LTRS, NZDL, CORA, and CORR. There are also many commercial services, one of the most well-known being the Science Citation Index (`http://www.isinet.com/`). The effectiveness of the available services varies according to discipline. The most successful free service appears to be the LANL e-Print archive [14], which has had great success in the physics community (a plan to shut down the service could not be carried out due to the response from the user community). These services are mostly complementary, providing different levels of comprehensiveness, recency, and features. None of the indices are comprehensive, so using multiple indices increases coverage, similar to using multiple web search engines [20, 21, 24].

3 CiteSeer

The CiteSeer project at NEC Research Institute [13] aims to improve the dissemination, retrieval, and accessibility of scientific literature. Specific areas of focus include the effective use of the capabilities of the web, and the use of

CIKM '99 11/99 Kansas City, MO, USA
© 1999 ACM 1-58113-146-1/99/0010...$5.00

machine learning. CiteSeer locates scientific articles on the web, extracts information such as the citations, citation context, article title, etc., and performs full-text indexing and autonomous citation indexing. Rather then providing just another digital library, CiteSeer provides algorithms, techniques, and software that can be used in other digital libraries. The project encompasses areas including:

1. *Location of articles.* The efficient location of scientific articles on the web.
2. *Full-text indexing* of articles, including the content of Postscript and PDF files.
3. *Autonomous Citation Indexing* (ACI) – the autonomous creation of an index of the links between scientific articles, similar to the Science Citation Index.
4. *Information extraction.* CiteSeer includes algorithms and machine learning techniques for automatically extracting information such as the title and author from indexed documents and individual citations.
5. *Query-sensitive summaries* of documents. Similar to the context display in the search engines Inquirus [19] and Google [6] (`http://www.google.com/`), CiteSeer aims to present results in such a way as to facilitate relevance estimation by the user, and improve the overall efficiency of the search process.
6. *Related documents.* CiteSeer employs new algorithms for the location of related documents, based on citation information in addition to the usual word information.
7. *Overlapping documents.* The web often contains minor revisions of articles, which lead to near duplicates in the database if not dealt with. CiteSeer identifies the amount of overlap between documents in order to detect these revisions.
8. *Citation graph analysis.* Analyzing the network of scientific literature. For example, similar to the work of Page et al. [22, 6] and Kleinberg [17], CiteSeer aims to identify "hubs" and "authorities" in the scientific literature.
9. *User profiling.* CiteSeer has a user profiling system which tracks the interests of users and recommends new citations and documents when they appear.
10. *Distributed error correction.* CiteSeer allows users of the system to correct errors in the database.
11. *External links.* Where possible, CiteSeer links to articles in external databases.
12. *Universal article access.* Access to details, statistics, and external links for articles using a standard article key.

The following sections provide more details on these areas of the CiteSeer project.

3.1 Locating Scientific Articles

There are a number of possible approaches for locating scientific articles on the web. Brute force search of the web is possible but time consuming. A more efficient technique might use machine learning algorithms (or heuristics) to direct the crawl [10, 23]. This can help to locate more documents earlier in the crawl, but does not guarantee finding all papers without crawling the entire web.

CiteSeer uses a more efficient approach to the location of articles on the publicly indexable web. Specifically, CiteSeer uses web search engines such as AltaVista to directly locate pages likely to contain research articles. Multiple queries are used to the search engines using various keywords likely to match pages containing scientific articles (e.g. "Postscript", "PDF", "technical report", "conference", "proceedings", etc.). Multiple search engines are used because this substantially improves the coverage of the web [20, 21]. This method has a number of advantages. The method avoids the duplication of work involved in crawling the web, and allows the combination of search engines which substantially improves coverage over any single crawler currently available. The method also allows easy customization of the database to specific areas of the literature by including appropriate keywords in the queries. Hundreds of thousands of scientific articles can be quickly and efficiently located using this technique.

CiteSeer also supports crawling functionality, designed to be used from start pages located similar to the techniques above. We are not yet using this functionality because of the success of the more directed approach. Additionally, CiteSeer monitors mailing lists, automatically indexing new documents as they are posted.

Once services like CiteSeer become common, we expect that many researchers will register their papers directly, making all of these techniques less necessary (thousands of papers have already been registered with CiteSeer).

3.2 Full-Text Indexing

CiteSeer includes full-text indexing of the entire content of articles, similar to the New Zealand Digital Library [31, 32]. Postscript and PDF documents are converted to text using pstotext (`http://www.research.digital.com/SRC/virtualpaper/pstotext.html`) from the Digital Virtual Paper project (`http://www.research.digital.com/SRC/virtualpaper/home.html`). The full-text indexing performed by CiteSeer is similar to the standard techniques [30], however there are some important differences.

CiteSeer aims to be as up to date as possible. Therefore the indexing is designed for continuous operation, so that updates can be performed continuously, without building or merging a new index. Index organization is similar to previous work [8, 11, 27]. CiteSeer maintains the usual hash table of words (inverted index) where each entry contains a compressed version of the word and a pointer to a block in a variable length record file that contains the matching documents and corresponding positions within the documents (compressed into a single bit stream with variable length identifiers). As the entries for each word grow the space allocated for them grows as a power of 2. CiteSeer supports full Boolean, phrase and proximity retrieval, using a standard recursive descent parser.

CiteSeer does not use any "stop" words (common words like "the", "a", etc. that are typically excluded from indexing). This is important for allowing higher precision search. One example where this is important is when searching for a specific author. Often author names are only specified in citations using initials instead of the full name. When looking for information on an author with a common last name it is important to be able to restrict the results only to those items that contain the correct first name or author initials. Thus, it is necessary to be able to search for phrases containing initials.

With a test database of about 200,000 documents and over 2.5 million citations, queries are typically executed in a fraction of a second on a Pentium Pro 200 machine. Performance degrades to about one second or longer when queries include phrases that contain very common words. In the demonstration database, queries using author initials (for example, `"m jordan" or "m i jordan"`) are common, accounting for about 20% of all queries. These queries were often taking several seconds due to the very frequent occurrence of initials in citations. In order to speed up these queries we cache the list of word positions and maintain a hash table for each initial indexed by document identifier (each entry contains the list of positions within the document). Proximity comparisons including initials typically do not need to examine the entire document list anymore. We randomly selected 25 queries that did not use initials and 25 that did in order to quantify the speedup. Table 1 shows the results. Without the speedup queries with initials were 7 times slower than other queries on average. With the speedup the queries with initials executed faster than the other queries.

Queries	Mean Execution Time
Without Initials	0.28s
With Initials	1.91s
With Initials and Hashing	0.26s

Table 1: Query execution time for 25 random queries with and without initials in a test database of about 200,000 documents containing over 2.5 million citations.

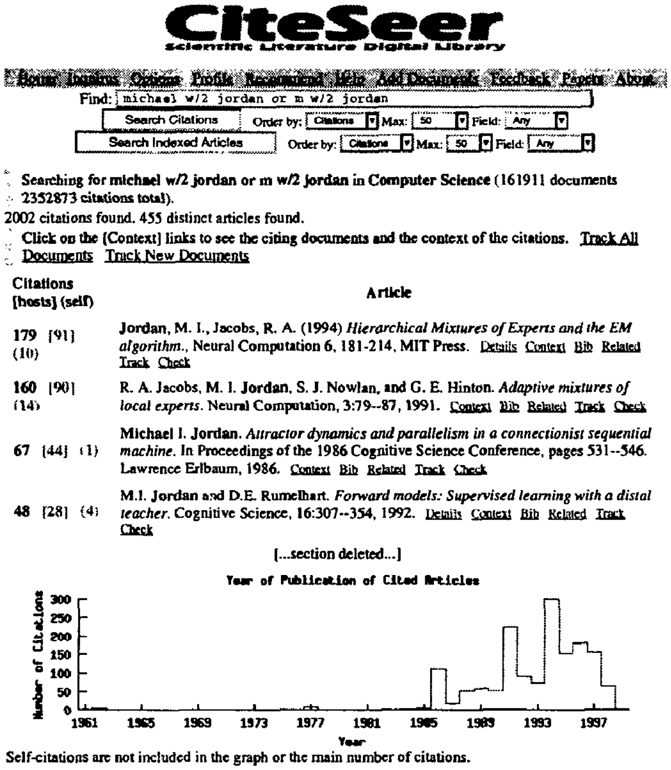
3.3 Autonomous Citation Indexing

CiteSeer includes autonomous citation indexing – the autonomous creation of a citation index similar to the Science Citation Index ® [12]. A citation index indexes the citations that research articles make, allowing, for example, the location of papers that cite a given paper. Autonomous citation indexing provides several advantages over traditional citation indexing. Traditional citation indexing requires manual effort. Automating the task as performed by CiteSeer should result in a reduction in cost and an increase in the availability of citation indices. An autonomous citation index can also provide more comprehensive and up-to-date indices of the literature – the Science Citation Index primarily indexes journal articles while CiteSeer can also index conference papers, preprints, technical reports, etc. The importance of indexing non-journal items varies by discipline, but is particularly important in areas like computer science where important research is often presented at conferences. We took a sample of 10 papers from the WWW7 conference and analyzed the distribution of references. We found that only 19.7% of references were to journal papers, while 30.3% were to conference papers, 18.0% were to books, and 32.0% were to technical reports, theses, and web pages.

For details of the citation matching and citation indexing in CiteSeer see [3, 13]. For related research, see the Open Journal Project [15], and Cameron's [9] proposal of a "universal, [Internet-based,] bibliographic and citation database linking every scholarly work ever written".

Note that CiteSeer has a general philosophy of investigating word-insensitive algorithms before introducing algorithms that use specific word information. This is in order to minimize bias in the errors made by the system. For example, it is simple to create a probabilistic model that labels the individual fields of citations by using the probability of each specific word belonging to certain fields. Such an algorithm can work well, however the algorithm depends critically on the coverage and recency of available training data, and errors are likely to be biased towards authors, titles, etc. that are not contained in the training data. This may correspond to a bias against new authors, new subjects, etc. that could potentially have a negative effect on scientific dissemination.

Figure 1 shows a sample CiteSeer response for a search within the citations extracted from articles. Citations to the same paper that may be written in different formats are grouped together [13]. Articles can be sorted according to the number of citations to them or by date. The "hosts" and "self" numbers indicate the number of distinct hosts that the citing articles were found on, and the number of citations predicted to be self-citations. The graph at the bottom shows the number of citations versus the year of publication of the cited articles. The "Context" links show the context of the citations, the "Bib" links provide a BibTeX entry for the article, the "Track" links activate tracking for the article (new citations will be emailed to the user), and the "Check" links display the individual citations that were grouped together as the same article (this can be used for detecting errors in the citation matching algorithms). The "Field" selection allows restricting the search results to the author or title fields.


CiteSeer
Scientific Literature Digital Library
Home Inquiries Options Profile Recommend Help Add Documents Feedback Papers About
Search Citations
Search Indexed Articles
Year of Publication of Cited Articles
Number of Citations
300
250
200
150
100
50
0
1961
1965
1969
1973
1977
1981
1985
1989
1993
1997
Year

Find: michael w/2 jordan or m w/2 jordan

Order by: Citations Max: 50 Field: Any

Order by: Citations Max: 50 Field: Any

Searching for michael w/2 jordan or m w/2 jordan in Computer Science (161911 documents 2352873 citations total).

2002 citations found. 455 distinct articles found.

Click on the [Context] links to see the citing documents and the context of the citations. Track All Documents Track New Documents

Citations [hosts] (self)	Article
179 [91] (10)	Jordan, M. I., Jacobs, R. A. (1994) *Hierarchical Mixtures of Experts and the EM algorithm.*, Neural Computation 6, 181-214, MIT Press. Details Context Bib Related Track Check
160 [90] (14)	R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. *Adaptive mixtures of local experts*. Neural Computation, 3:79--87, 1991. Context Bib Related Track Check
67 [44] (1)	Michael I. Jordan. *Attractor dynamics and parallelism in a connectionist sequential machine*. In Proceedings of the 1986 Cognitive Science Conference, pages 531--546. Lawrence Erlbaum, 1986. Context Bib Related Track Check
48 [28] (4)	M.I. Jordan and D.E. Rumelhart. *Forward models: Supervised learning with a distal teacher*. Cognitive Science, 16:307--354, 1992. Details Context Bib Related Track Check

[...section deleted...]

Self-citations are not included in the graph or the main number of citations.

Figure 1: Sample response of CiteSeer for a search within the citations. Articles can be sorted according to the number of citations to them or by date. The "hosts" and "self" numbers indicate the number of distinct hosts that the citing articles were found on, and the number of citations predicted to be self-citations. The graph at the bottom shows the number of citations versus the year of publication of the cited articles. The "Context" links show the context of the citations, the "Bib" links provide a BibTeX entry for the article, the "Track" links activate tracking for the article (new citations will be emailed to the user), and the "Check" links display the individual citations that were grouped together as the same article. Article titles are automatically highlighted. The syntax w/2 in the query means that words must appear within a maximum distance of 2 words.

3.4 Information Extraction

CiteSeer performs several types of information extraction on the indexed documents:

1. Extraction of the list of citations. CiteSeer can identify the citation list in a document, re-order documents that print in reverse order, remove page numbers from within the citation list, and delineate individual citations using citation tags, vertical spacing or indentation.
2. Extraction of the context of citations. For each citation made in a document, CiteSeer can extract the context of the article where the citation is made. Regular expressions are used to identify the references in the text which may differ from the citation tag used in the citation list.
3. Extraction of subfields from citations. CiteSeer extracts fields such as the title and author from citations. For more details see [13].
4. Extraction of bibliographic details of the articles being indexed. CiteSeer can identify the indexed articles by extracting the title and author from the header of the document. This is simple to do with reasonably high accuracy by analyzing the font information. The algorithm currently used by CiteSeer is as follows:

   – The document is marked up with tags indicating font changes. Each font change is identified by the average width of characters contained in the font.

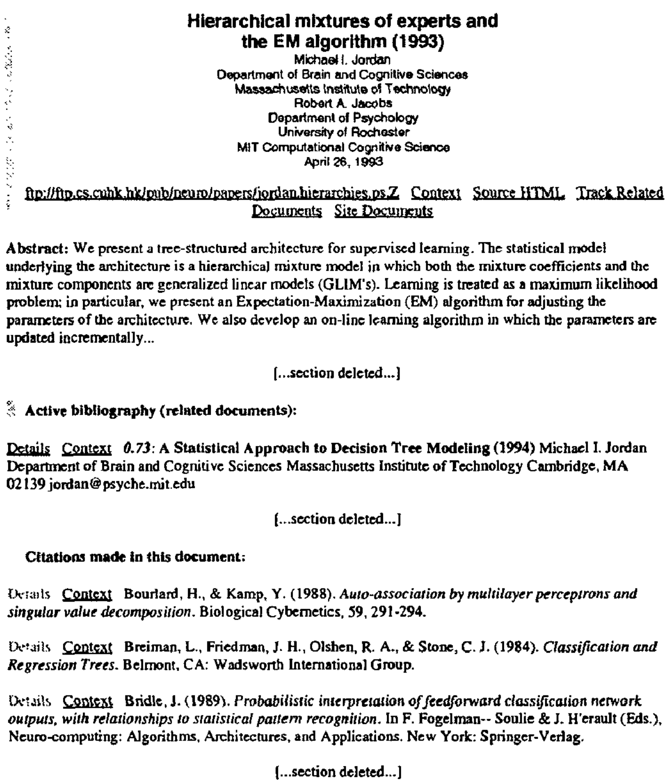
   – Font markup is changed to reflect relative font changes with respect to the most common font size in the document.

   – Heuristics search for the title of the document within the resulting representation (for example, the title is often written in the largest font in the header of the document).

For a hidden Markov model approach to extracting subfields see [25]. Figure 2 shows a sample of the details available for each document indexed by CiteSeer. The document header, abstract, and citations can be seen, along with an "active bibliography" of related documents.

3.5 Context and Query-Sensitive Summaries

Once a user locates an article of possible interest in the database, CiteSeer can display the context of how that article is cited in subsequent publications. When searching within the indexed documents, CiteSeer displays sample context of the query terms within the documents. These techniques typically help the user to more efficiently determine the relevance of the documents in question. In general, query-sensitive summaries of documents have been



Hierarchical mixtures of experts and
the EM algorithm (1993)
Michael I. Jordan
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Robert A. Jacobs
Department of Psychology
University of Rochester
MIT Computational Cognitive Science
April 26, 1993

ftp://ftp.cs.cuhk.hk/pub/neuro/papers/jordan.hierarchies.ps.Z Context Source HTML Track Related Documents Site Documents

Abstract: We present a tree-structured architecture for supervised learning. The statistical model underlying the architecture is a hierarchical mixture model in which both the mixture coefficients and the mixture components are generalized linear models (GLIM's). Learning is treated as a maximum likelihood problem; in particular, we present an Expectation-Maximization (EM) algorithm for adjusting the parameters of the architecture. We also develop an on-line learning algorithm in which the parameters are updated incrementally...

[...section deleted...]

Active bibliography (related documents):

Details Context *0.73*: **A Statistical Approach to Decision Tree Modeling (1994)** Michael I. Jordan Department of Brain and Cognitive Sciences Massachusetts Institute of Technology Cambridge, MA 02139 jordan@psyche.mit.edu

[...section deleted...]

Citations made in this document:

Details Context Bourlard, H., & Kamp, Y. (1988). *Auto-association by multilayer perceptrons and singular value decomposition*. Biological Cybernetics, 59, 291-294.

Details Context Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

Details Context Bridle, J. (1989). *Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition*. In F. Fogelman-- Soulie & J. H'erault (Eds.), Neuro-computing: Algorithms, Architectures, and Applications. New York: Springer-Verlag.

[...section deleted...]

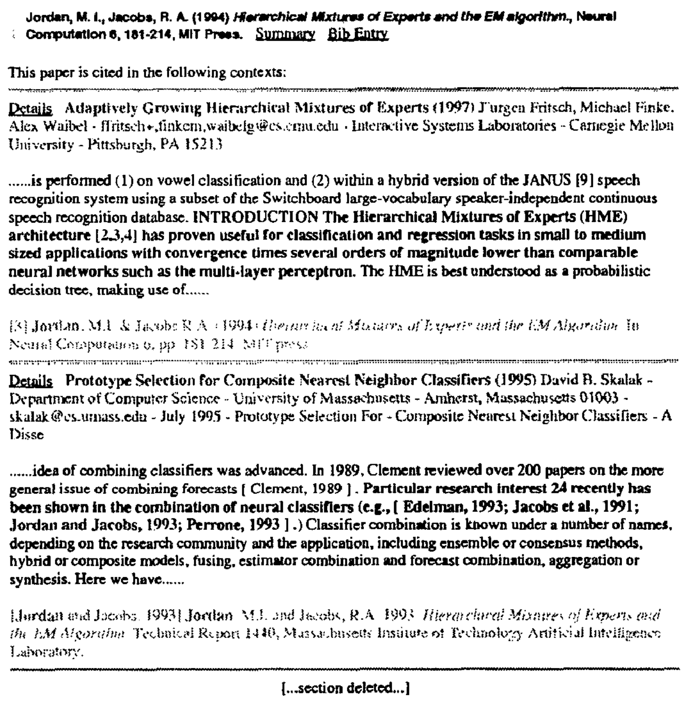
Figure 2: Sample detail information for an article in the CiteSeer database. The document header, abstract, and citations can be seen, along with an "active bibliography" of related documents.

shown to improve search efficiency [28, 29]. Tombros performed a user study that showed that users working with the query-sensitive summaries had a higher success rate. The query-sensitive summaries allowed the users to more accurately and rapidly determine the relevance of documents, and greatly reduced the need to refer to the full text of documents. Figure 3 shows sample citation context and figure 4 shows a sample of the response of CiteSeer for a search within the text of the indexed articles.

3.6 Related Documents

Research articles contain citations to related and previous research. CiteSeer augments the citation list of articles by locating other related documents using algorithms based on word and citation information. Details of the algorithms can be found in [13]. When viewing the details of a document, CiteSeer displays an "active bibliography", showing the most related documents. The active bibliography is computed in a fraction of a second in real-time, and changes to reflect changes to the database as new documents are indexed.

One observation we have made regarding the active bibliographies is that papers by the same authors or authors at the same institution are often ranked highly, as might be expected. It may therefore be useful to separately identify such articles in order to highlight related documents from other authors.



Jordan, M. I., Jacobs, R. A. (1994) ***Hierarchical Mixtures of Experts and the EM algorithm.*****, Neural Computation 6, 181-214, MIT Press.** Summary Bib Entry

This paper is cited in the following contexts:

Details Adaptively Growing Hierarchical Mixtures of Experts (1997) Jurgen Fritsch, Michael Finke. Alex Waibel - ffritsch+,finkem,waibelg@cs.cmu.edu - Interactive Systems Laboratories - Carnegie Mellon University - Pittsburgh, PA 15213

......is performed (1) on vowel classification and (2) within a hybrid version of the JANUS [9] speech recognition system using a subset of the Switchboard large-vocabulary speaker-independent continuous speech recognition database. **INTRODUCTION The Hierarchical Mixtures of Experts (HME) architecture [2,3,4] has proven useful for classification and regression tasks in small to medium sized applications with convergence times several orders of magnitude lower than comparable neural networks such as the multi-layer perceptron.** The HME is best understood as a probabilistic decision tree, making use of......

[3] Jordan, M.I. & Jacobs R.A. (1994) *Hierarchical Mixtures of Experts and the EM Algorithm.* In Neural Computation 6, pp. 181-214. MIT press.

Details **Prototype Selection for Composite Nearest Neighbor Classifiers (1995)** David B. Skalak - Department of Computer Science - University of Massachusetts - Amherst, Massachusetts 01003 - skalak@cs.umass.edu - July 1995 - Prototype Selection For - Composite Nearest Neighbor Classifiers - A Disse

......idea of combining classifiers was advanced. In 1989, Clement reviewed over 200 papers on the more general issue of combining forecasts [Clement, 1989] . **Particular research interest 24 recently has been shown in the combination of neural classifiers (e.g., [Edelman, 1993; Jacobs et al., 1991; Jordan and Jacobs, 1993; Perrone, 1993] .)** Classifier combination is known under a number of names, depending on the research community and the application, including ensemble or consensus methods, hybrid or composite models, fusing, estimator combination and forecast combination, aggregation or synthesis. Here we have......

[Jordan and Jacobs, 1993] Jordan, M.I. and Jacobs, R.A. 1993 *Hierarchical Mixtures of Experts and the EM Algorithm.* Technical Report 1440, Massachusetts Institute of Technology Artificial Intelligence Laboratory.

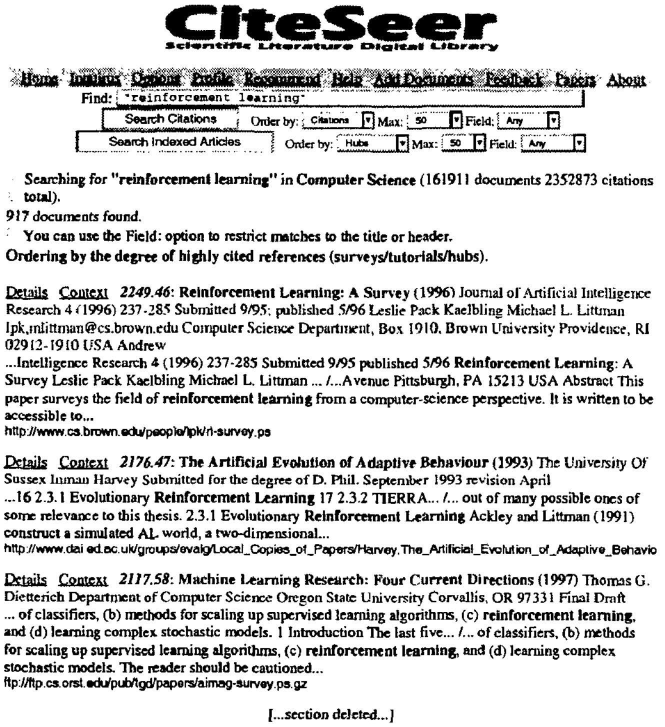
[...section deleted...]

Figure 3: Sample citation context information for an article in the CiteSeer database. For each article citing the given article of interest, the header, context of the citation, and the specific form of the citation can be seen. Note that the two citations grouped together above actually refer to a technical report and the corresponding journal article. This is by design, the algorithm currently used in the demonstration system groups together articles with the same title and authors.

3.7 Overlapping Documents

There are many duplicate research articles on the web. Identical documents are easy to detect (CiteSeer uses SHA checksums), however there are many minor revisions of articles that would lead to duplicates in the digital library unless detected. For example, two co-authors might have the same article online but one of the authors might have made a minor revision to the article (e.g. by adding the publication details).

CiteSeer takes a sentence based approach to detecting these revisions. A database of all sentences is maintained and the percentage of identical sentences is computed between all documents. Pairs of documents with a very high percentage


CiteSeer
Scientific Literature Digital Library
Home Indexes Options Profile Recommend Help Add Documents Feedback Papers About
Search Citations
Search Indexed Articles

Find: "reinforcement learning"

Order by: Citations Max: 50 Field: Any

Order by: Hubs Max: 50 Field: Any

Searching for "reinforcement learning" in Computer Science (161911 documents 2352873 citations **total).**

917 documents found.

You can use the Field: option to restrict matches to the title or header.

Ordering by the degree of highly cited references (surveys/tutorials/hubs).

Details Context *2249.46*: **Reinforcement Learning: A Survey** (1996) Journal of Artificial Intelligence Research 4 (1996) 237-285 Submitted 9/95; published 5/96 Leslie Pack Kaelbling Michael L. Littman lpk,mlittman@cs.brown.edu Computer Science Department, Box 1910, Brown University Providence, RI 02912-1910 USA Andrew
...Intelligence Research 4 (1996) 237-285 Submitted 9/95 published 5/96 **Reinforcement Learning**: A Survey Leslie Pack Kaelbling Michael L. Littman ... /...Avenue Pittsburgh, PA 15213 USA Abstract This paper surveys the field of **reinforcement learning** from a computer-science perspective. It is written to be accessible to...
http://www.cs.brown.edu/people/lpk/rl-survey.ps

Details Context *2176.47*: **The Artificial Evolution of Adaptive Behaviour** (1993) The University Of Sussex Inman Harvey Submitted for the degree of D. Phil. September 1993 revision April
...16 2.3.1 Evolutionary **Reinforcement Learning** 17 2.3.2 TIERRA... /... out of many possible ones of some relevance to this thesis. 2.3.1 Evolutionary **Reinforcement Learning** Ackley and Littman (1991) construct a simulated AL world, a two-dimensional...
http://www.dai.ed.ac.uk/groups/evalg/Local_Copies_of_Papers/Harvey.The_Artificial_Evolution_of_Adaptive_Behavio

Details Context *2117.58*: **Machine Learning Research: Four Current Directions** (1997) Thomas G. Dietterich Department of Computer Science Oregon State University Corvallis, OR 97331 Final Draft
... of classifiers, (b) methods for scaling up supervised learning algorithms, (c) **reinforcement learning**, and (d) learning complex stochastic models. 1 Introduction The last five... /... of classifiers, (b) methods for scaling up supervised learning algorithms, (c) **reinforcement learning**, and (d) learning complex stochastic models. The reader should be cautioned...
ftp://ftp.cs.orst.edu/pub/tgd/papers/aimag-survey.ps.gz

[...section deleted...]

Figure 4: Sample response of CiteSeer for a search within the full-text of indexed articles, ordered by "hubs". Articles can be sorted according to citations, "hubs", and date. Query-sensitive summaries are shown for each article highlighting the context of the query terms within the document. The search can be restricted to the title or header fields using the "Field" selection.

of identical sentences are considered duplicates. Sentence identification is non-trivial, however correct identification of sentences is not necessary for this application. CiteSeer simply uses punctuation to delineate sentences (periods, exclamation marks, and question marks), although this is not always correct (e.g. consider abbreviations).

Sentences are stored in a hash table, where each entry contains a list of the documents containing the sentence. Sentences are preprocessed by removing non-alphabetic characters and vowels, truncating to a maximum length (currently 80), ignoring short sentences (currently those with less than 30 characters), and folding 5 additional characters onto other characters to enable packing two characters per byte. In order to avoid the 2Gb maximum file size limitation imposed by some operating systems, entries are split between multiple files. Sentences in the header and citation list of documents are ignored.

A similar sentence based technique is used by COPS to detect copies of documents [5]. Another well-known copy detection mechanism is SCAM [26], which uses word frequencies and works better for detecting documents with partial overlap. Also related is [7], however the algorithm they present is more expensive than necessary for the application considered here.

3.8 Citation Graph Analysis

CiteSeer currently performs two different types of analysis on the graph formed by citation links. CiteSeer predicts whether or not citations are self-citations by comparing the authors in the citations with the authors in the header of the indexed articles. This information is used when ranking documents based on citations (currently, self-citations are not counted).

Page et al. [22] and Kleinberg [17] have introduced methods of ranking web pages using link information (also see the improvements to Kleinberg's algorithm by Bharat and Henzinger [2]). Kleinberg's algorithm identifies "hubs" and "authorities". "Authorities" are pages with many links to them, while "hubs" are pages containing many links to authorities. In the context of scientific articles, we suggest that "hub" articles can be useful for an introduction to a field, and correspond to review, survey, or tutorial style articles. Review articles that summarize important literature are relatively rare in some areas, however the ranking of regular articles as hubs may be useful because these papers can contain good introductions to areas of the literature in their prior work and introductory sections. CiteSeer currently allows ordering articles by either "hubs" or "authorities". We are investigating extensions to the Page/Kleinberg algorithms, such as normalization according to the number of citations contained in articles, and incorporation of the temporal aspect of citations (more recent articles are expected to have fewer citations). CiteSeer can order results based on the number of citations compared to the expected number of citations, considering the time since the article was published.

We are also interested in analyzing the graph of citations to identify communities and experts (see [16]), and to analyze the relationships between articles and the evolution of the literature. One interesting application of citation graph analysis and/or related document algorithms is the identification of potential reviewers for a given article. This may help to locate more appropriate reviewers for given articles. (Currently, some editors use web search engines to help them locate potential reviewers).

3.9 User Profiling

CiteSeer has a system for automatically recommending new relevant documents via email or the web interface [4]. For example, new documents that contain specific keywords or citations, or that are related to specific documents can be recommended. A personal profile is maintained for each user. The profile can be updated manually by the user, or automatically with machine learning based on browsing patterns or responses to recommendations. Users can remain anonymous in which case they are identified only by a unique identification number stored in a HTTP cookie.

3.10 Distributed Error Correction

CiteSeer uses *distributed error correction* to allow individual users to correct errors in the database. See [18] for details of the implementation and issues such as trust, recovery, detecting malicious changes, and the use of correction information to improve automated algorithms or predict the probability of errors.

3.11 External Links

Some papers are not freely available on the web due to copyright restrictions, however they may be available in publisher databases. When possible, CiteSeer links citations in the database to external publishers. The ease of this task depends on the organization of the individual databases. The American Physical Society (APS) provides a good example and allows easy linking of papers. A sample URL for a paper in the APS database is: `http://publish.aps.org/abstract/PRD/v10/p20`. This link would refer to a paper in volume 10 of Physical Review D on page 20. It is relatively simple for CiteSeer to extract this information from citations. CiteSeer currently generates these links in realtime.

3.12 Universal Article Access

CiteSeer contains many kinds of information about articles. For source articles, CiteSeer has many details including the title, authors, abstract, citations, and full-text. However, CiteSeer has citation details and statistics for all articles cited by any of the source articles. CiteSeer also knows how to find articles in selected external databases. CiteSeer allows access to all of the information for a given article using a universal article key. Currently, this key consists of the last name of the first author of the article, the year of publication, and the first word of the article title (ignoring "the", "a", etc.), although alternative keys are likely to be supported in the future. This key is unique for a large percentage of articles, but not for all articles. When the key is not unique CiteSeer presents all articles with the same key for user selection.

4 Availability

Perhaps most importantly, NEC Research Institute has made the software for CiteSeer available at no cost for non-commercial use. To obtain the latest version contact `citeseer@research.nj.nec.com`. There is a mailing list for CiteSeer announcements, to join the list send a message to `majordomo@research.nj.nec.com` with `subscribe citeseer-announce` in the body of the message. A demonstration CiteSeer service indexing over 200,000 computer science articles containing over 3 million citations can be found at `http://csindex.com/`.

5 Summary

CiteSeer is a digital library that aims to improve the dissemination, retrieval, and accessibility of scientific literature on the web. Specific areas of focus include the effective use of the capabilities of the web, and the use of machine learning. Software and data from the CiteSeer project is available at no cost for non-commercial use, which we hope will encourage extensions of this and related work.

References

[1] J.M. Barrie and D.E. Presti. The World Wide Web as an instructional tool. *Science*, 274:371–372, 1996.

[2] K. Bharat and M.R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[3] Kurt Bollacker, Steve Lawrence, and C. Lee Giles. CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications. In Katia P. Sycara and Michael Wooldridge, editors, *Proceedings of the Second International Conference on Autonomous Agents*, pages 116–123, New York, 1998. ACM Press.

[4] Kurt Bollacker, Steve Lawrence, and C. Lee Giles. A system for automatic personalized tracking of scientific literature on the web. In *Digital Libraries 99 - The Fourth ACM Conference on Digital Libraries*, pages 105–113, New York, 1999. ACM Press.

[5] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Proceedings of the ACM SIGMOD Annual Conference*, 1995.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World Wide Web Conference*, Brisbane, Australia, 1998.

[7] Andrei Broder, Steve Glassman, Mark Manasse, and Geoffrey Zweig. Syntactic clustering of the web. In *Sixth International World Wide Web Conference*, pages 391–404, 1997.

[8] Eric W. Brown, James P. Callan, and W. Bruce Croft. Fast incremental indexing for full-text information retrieval. In *Proceedings of the 20th International Conference on Very Large Databases*, pages 192–202, 1994.

[9] Robert D. Cameron. A universal citation database as a catalyst for reform in scholarly communication. *First Monday*, 2(4), 1997.

[10] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through URL ordering. In *Proceedings of the Seventh World-Wide Web Conference*, 1998.

[11] Doug Cutting and Jan Pedersen. Optimizations for dynamic inverted index maintenance. In *Proceedings of the 13th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 405–411, 1990.

[12] Eugene Garfield. *Citation Indexing: Its Theory and Application in Science, Technology, and Humanities.* Wiley, New York, 1979.

[13] C. Lee Giles, Kurt Bollacker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In Ian Witten, Rob Akscyn, and Frank M. Shipman III, editors, *Digital Libraries 98 - The Third ACM Conference on Digital Libraries*, pages 89–98, Pittsburgh, PA, June 23–26 1998. ACM Press.

[14] P. Ginsparg. First steps towards electronic research communication. *Computers in Physics*, 8:390–396, 1994.

[15] S. Hitchcock, L. Carr, S. Harris, J.M.N. Hey, and W. Hall. Citation linking: Improving access to online journals. In Robert B. Allen and Edie Rasmussen, editors, *Proceedings of the 2nd ACM International Conference on Digital Libraries*, pages 115–122, New York, NY, 1997. ACM.

[16] H. Kautz, B. Selman, and M. Shah. ReferralWeb: Combining social networks and collaborative filtering. *Communications of the ACM*, 30(3), 1997.

[17] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, San Francisco, California, 25–27 January 1998.

[18] Steve Lawrence, Kurt Bollacker, and C. Lee Giles. Distributed error correction. In *Digital Libraries 99 - The Fourth ACM Conference on Digital Libraries*, page 232, New York, 1999. ACM Press.

[19] Steve Lawrence and C. Lee Giles. Context and page analysis for improved web search. *IEEE Internet Computing*, 2(4):38–46, 1998.

[20] Steve Lawrence and C. Lee Giles. Searching the World Wide Web. *Science*, 280(5360):98–100, 1998.

[21] Steve Lawrence and C. Lee Giles. Accessibility of information on the web. *Nature*, 400(6740):107–109, 1999.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. 1998.

[23] J. Rennie and A. McCallum. Using reinforcement learning to spider the web efficiently. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, 1999.

[24] E. Selberg and O. Etzioni. Multi-service search and comparison using the MetaCrawler. In *Proceedings of the 1995 World Wide Web Conference*, 1995.

[25] Kristie Seymore, Andrew McCallum, and Roni Rosenfeld. Learning hidden Markov model structure for information extraction. In *AAAI 99 Workshop on Machine Learning for Information Extraction*, 1999.

[26] N. Shivakumar and H. Garcia-Molina. SCAM: A copy detection mechanism for digital documents. In *2nd International Conference on the Theory and Practice of Digital Libraries*, 1995.

[27] Anthony Tomasic, Hector Garcia-Molina, and Kurt Shoens. Incremental updates of inverted lists for text document retrieval. In *Proceedings of the 1994 ACM SIGMOD Conference*, pages 289–300, 1994.

[28] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of SIGIR 98*, Melbourne, Australia, 1998.

[29] Anastasios Tombros. *Reflecting User Information Needs Through Query Biased Summaries.* PhD thesis, Department of Computer Science, University of Glasgow, September 1997.

[30] I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and indexing documents and images.* Van Nostrand Reinhold, New York, NY, 1994.

[31] I.H. Witten, C.G. Nevill-Manning, and S.J.Cunningham. Building a digital library for computer science research: technical issues. In *Proceedings Australasian Computer Science Conference*, Melbourne, Australia, January 1996.

[32] I.H. Witten, C.G. Nevill-Manning, and S.J.Cunningham. Digital libraries based on full-text retrieval. In *Proceedings of WebNet 96*, San Francisco, October 1996.