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
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 than 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.

<table>
<thead>
<tr>
<th>Queries</th>
<th>Mean Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Initials</td>
<td>0.28s</td>
</tr>
<tr>
<td>With Initials</td>
<td>1.91s</td>
</tr>
<tr>
<td>With Initials and Hashing</td>
<td>0.26s</td>
</tr>
</tbody>
</table>

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.

Citation Index® [12]. A citation 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

### 3.3 Autonomous Citation Indexing

CiteSeer includes autonomous citation indexing – the autonomous creation of a citation index similar to the Science Citation Index®. The autonomous citation indexing involves the following steps:

1. **Citation Extraction**: Extract citations from the input documents.
2. **Preprocessing**: Preprocess the citations to remove noise and standardize formats.
3. **Pattern Recognition**: Use pattern recognition techniques to identify citation patterns.
4. **Indexing**: Use indexing techniques to create a citation index.
5. **Citation Counting**: Count the citations for each paper.
6. **Result Presentation**: Present the results in a user-friendly format.

This approach allows CiteSeer to index a wide range of documents, including conference papers, technical reports, and web pages, without the need for manual labor. The autonomous nature of the indexing process also ensures that the index is up-to-date and comprehensive.

### 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.

<table>
<thead>
<tr>
<th>Queries</th>
<th>Mean Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Initials</td>
<td>0.28s</td>
</tr>
<tr>
<td>With Initials</td>
<td>1.91s</td>
</tr>
<tr>
<td>With Initials and Hashing</td>
<td>0.26s</td>
</tr>
</tbody>
</table>

Citation Index® [12]. A citation 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.

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 Cognition Science
April 20, 1993

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 (GLMs). 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.

Active bibliography (related documents):

Citations made in this document:

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.

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 citations grouped together above actually refer to a technical report or the corresponding journal article.

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 among these 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 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 rec-
ommended. 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 real-time.

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 cite@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 cite-seer-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


