Accepted for publication, Digital Libraries '98 - The Third ACM Conference on Digital Libraries, Pittsburgh, PA, USA, June 23-26, 1998.

CiteSeer: An Automatic Citation Indexing System

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

ABSTRACT

We present *CiteSeer*: an autonomous citation indexing system which indexes academic literature in electronic format (e.g. Postscript files on the Web). CiteSeer understands how to parse citations, identify citations to the same paper in different formats, and identify the context of citations in the body of articles. CiteSeer provides most of the advantages of traditional (manually constructed) citation indexes (e.g. the ISI citation indexes), including: literature retrieval by following citation links (e.g. by providing a list of papers that cite a given paper), the evaluation and ranking of papers, authors, journals, etc. based on the number of citations, and the identification of research trends. CiteSeer has many advantages over traditional citation indexes, including the ability to create more up-to-date databases which are not limited to a preselected set of journals or restricted by journal publication delays, completely autonomous operation with a corresponding reduction in cost, and powerful interactive browsing of the literature using the context of citations. Given a particular paper of interest, CiteSeer can display the context of how the paper is cited in subsequent publications. This context may contain a brief summary of the paper, another author's response to the paper, or subsequent work which builds upon the original article. CiteSeer allows the location of papers by keyword search or by citation links. Papers related to a given paper can be located using common citation information or word vector similarity. CiteSeer will soon be available for public use.

KEYWORDS: citation indexing, citation context, literature search, bibliometrics.

INTRODUCTION

A citation index [6] indexes the links between articles that researchers make when they cite other articles. Citation indexes are very useful for a number of purposes, including literature search, evaluation, and analysis of the academic literature. This paper introduces *CiteSeer*, which is an automatic citation indexing system. CiteSeer provides most of the advantages of traditional (manually constructed) citation indexes (e.g. the ISI citation indexes [10]), including: literature retrieval by following citation links (e.g. by providing a list of papers that cite a given paper), the evaluation and ranking of papers, authors, journals, etc. based on the number of citations, and the identification of research trends. CiteSeer has many advantages over traditional citation indexes, including a more up-to-date database which is not limited to a preselected set of journals or restricted by journal publication delays, completely autonomous operation with a corresponding reduction in cost, and powerful interactive browsing of the literature using the context of citations.

CITATION INDEXING

References contained in academic articles are used to give credit to previous work in the literature and provide a link between the "citing" and "cited" articles. A citation index [6] indexes the citations that an article makes, linking the articles with the cited works. Citation indexes were originally designed mainly for information retrieval [7]. The citation links allow navigating the literature in unique ways. Papers can be located independent of language, and words in the title, keywords or document. A citation index allows navigation backward in time (the list of cited articles) and forward in time (which subsequent articles cite the current article?) Citation indexes can be used in many ways, e.g. a) citations can help to find other publications which may be of interest, b) the context of citations in citing publications may be helpful in judging the important contributions of a cited paper and the usefulness of a paper for a given query [7, 14], c) a citation index allows finding out where and how often a particular article is cited in the literature, thus providing an indication of the importance of the article, and d) a citation index can provide detailed analyses of research trends and identify emerging areas of science [8].

Existing Citation Indexes

The Institute for Scientific Information (ISI) [10] produces multidisciplinary citation indexes. One is the *Science Citation Index* ® (SCI), intended to be a practical, cost-effective tool for indexing the significant scientific journals. The ISI databases are valuable and useful tools. A recurrent criticism against the ISI databases is that they are biased because of the management decisions of ISI, with respect to the selection of the items which are indexed [5]. Other ISI services include Keywords Plus ® [7], which adds citation information to the indexing of an article, *Research Alert* ®, which provides weekly listings of citations related to a set of key references [8], and *bibliographic coupling*, which allows navigation by locating papers which share one or more references [8].

A Universal Citation Database

Cameron proposed a universal bibliographic and citation database which would link every scholarly work ever written [3]. He describes a system in which all published research would be available to and searchable by any scholar with Internet access. The database would include citation links and would be comprehensive and up-to-date. Perhaps the most important difference between Cameron's vision of a universal citation database and CiteSeer is that CiteSeer does not require any extra effort on the part of authors beyond placement of their work on the Web. CiteSeer automatically creates the citation database from downloaded documents whereas Cameron has proposed a system which requires authors or institutions to provide citation information in a specific format. A second relevant difference is that CiteSeer exists whereas Cameron's system is only a proposal that presents significant difficulty for implementation. Additionally, CiteSeer can extract the context of citations, improving literature search and evaluation.

CITESEER

CiteSeer downloads papers that are made available on the World Wide Web, converts the papers to text, parses them to extract the citations and the context in which the citations are made in the body of the paper, and stores the information in a database. CiteSeer includes algorithms for identifying and grouping variant forms of citations to the same paper. CiteSeer also performs full-text indexing of the articles and citations as well as providing support for browsing via citation links. Papers related to a given paper can be located using common citation information or word vector similarity. Given a particular paper of interest, CiteSeer can display the context of how the paper is cited in subsequent publications. This context may contain a brief summary of the paper, another author's response to the paper, or subsequent work which builds upon the original article.

Compared to the current commercial citation indexes, the citation indexing performed by CiteSeer has the following disadvantages:

1. CiteSeer does not cover the significant journals as comprehensively. We expect that this will be less of a disadvantage over time as more journals become available online (agreements with the publishers would be required to index most journals).
2. CiteSeer cannot distinguish subfields as accurately, e.g. CiteSeer will not disambiguate two authors with the same name. We expect that CiteSeer will improve in this regard over time due to the collection of databases and improvement of the algorithms used in CiteSeer.

The advantages of CiteSeer compared to the current commercial citation indexes include:

1. Because CiteSeer can index articles as soon as they are available on the Web, it should be of greater use to researchers for finding recent relevant literature, and for keeping up to date.
2. CiteSeer is autonomous, requiring no manual effort during indexing.
3. CiteSeer can be used to make a more informed estimation of the impact of a given article (citations do not always imply scholarly impact [2, 4, 9, 12, 17], and CiteSeer helps by making the context of citations easily and quickly browsable).

The following sections describe the document acquisition, document parsing, identification of identical citations, and database query/browsing.

Document Acquisition

CiteSeer can be used to create a comprehensive index of literature on the Web, or to create indexes of a user-specified topic. There are many ways in which CiteSeer can locate papers to index, e.g. a Web crawler could be used similar to the Web search engines, the location information could be extracted from the announcements of papers in Usenet message groups or mailing lists, or CiteSeer might index new issues of journals as they are made available (under agreement with the publisher). Currently, CiteSeer uses Web search engines (e.g. AltaVista, HotBot, Excite) and heuristics to locate papers (e.g. CiteSeer can search for pages which contain the words "publications", "papers", "postscript", etc.). CiteSeer locates and downloads Postscript files identified by ".ps", ".ps.Z", or ".ps.gz" extensions. URLs and Postscript files that are duplicates of those already found are detected and skipped.

Document Parsing

The downloaded Postscript files are first converted into text. We currently use PreScript from the New Zealand Digital Library project (`http://www.nzdl.org/technology/prescript.html`). The text file is checked to verify that the document is a valid research document by testing for the existence of a references or bibliography section. CiteSeer detects and reorders Postscript files that have their pages in reverse order. The following information is extracted from the documents:

- *URL:* The URL of the downloaded Postscript file is stored.
- *Header:* The title and author block of the paper is extracted.
- *Abstract:* If it exists, the abstract text is extracted.
- *Introduction:* If it exists, the introduction section is extracted.

- *Citations:* The list of references made by the document are extracted and parsed further as described below.
- *Citation context:* The context in which the document makes the citations is extracted from the body of the document.
- *Full text:* The full text of the document and citations is indexed.

Due to the wide variation in the formatting of document headers, not all subfields (title, author, author affiliation, addresses, etc.) are reliably detected. We plan to improve the parsing of document headers using either additional heuristics or machine learning techniques.

Once the set of references has been identified, individual citations are extracted. Each citation is parsed using heuristics to extract the following fields: title, author, year of publication, page numbers, and citation tag. The citation tag is the information in the citation that is used to cite that article in the body of the document (e.g. “[6]”, “[Giles97]”, “Marr 1982”). The citation tags are used to find the locations in the document body where the citations are actually made, allowing us to extract the context of these citations.

The heuristics used to parse the citations were constructed using an “invariants first” philosophy. That is, subfields of a citation which had relatively uniform syntactic indicators as to their position and composition given all previous parsing, were always parsed next. For example, the label of a citation to mark it in context always exists at the beginning of a citation and the format is uniform across all citations. Once the more regular features of a citation were identified, trends in syntactic relationships between subfields to be identified and those already identified were used to predict where the desired subfield existed (if at all). For example, author information almost always precedes title information, and publisher almost always is after the title. We also use databases of author names, journal names, etc. which are used to help identify the subfields of the citations.

Parsing of natural language citations is difficult [1]. However, we have been able to achieve reasonably good results using heuristics to extract certain subfields. Table 1 shows parsing statistics for a sample CiteSeer database which was created from 5093 documents related to “neural networks”. We plan on using learning techniques and additional heuristics in order to extract additional fields of the citations. Note that we have not attempted to exhaustively index all papers found on the Web in this area. We can see that the titles and authors can be found in citations roughly 80% of the time and page numbers roughly 40% of the time. The low number for page numbers detected is probably due (at least partially) to the fact that many citations did not contain page numbers.

Database	*neural networks*
Documents parsed	5093
Citations found	89614
Titles identified	71908/89614 = 80.2%
Authors identified	73539/89614 = 82.1%
Page numbers identified	39595/89614 = 44.2%

Table 1: Citation parsing performance for a sample CiteSeer database.

Identifying Citations to the Same Article

Citations to a given article can be made in significantly different ways. For example, the following citations, extracted from neural network publications, are all to the same article:

```
[7] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J.
      Stone. Classification and Regression Trees.
      Wadsworth, Pacific Grove, California, 1984.
6. L. Breiman, J. Friedman, R. Olshen and C. Stone,
      Classification and Regression Trees, Wadsworth
      and Brooks, 1984.
[1] L. Breiman et al. Classification and Regression
      Trees. Wadsworth, 1984.
```

Much of the significance of CiteSeer derives from the ability to recognize that all of these citations refer to the same article. This ability allows:

1. A detailed listing of a cited article to show all instances of the citation across multiple articles.
2. Statistics on citation frequency to be generated, allowing the estimation of the importance of articles.
3. More accurate identification of sub-fields for a citation, e.g. errors or incomplete fields in one instance of a citation may be resolved from analysis of the group of citations to the same article. More accurate identification of sub-fields for a citation leads to more accurate estimation of statistics based on fields, e.g. the citation frequency for authors.

As suggested by the example citations above, the problem is not completely trivial. We have investigated three methods of identifying citations to identical articles and grouping them together, along with a simple baseline method for comparison. For all of these methods, we found that normalizing certain aspects of citations tends to improve the results. We have used the following normalizations:

1. Conversion to lowercase.
2. Removal of hyphens.
3. Removal of citation tags, e.g. [3], [Giles 92] at the beginning of the citation.
4. Expansion of common abbreviations, e.g. conf. → conference, proc. → proceedings. Common abbreviations for the following words are currently expanded: conference, proceedings, international, society, transactions, and technical report.

5. Removal of extraneous words and characters which may occur in some, but usually not all, instances of a citation. The following words are removed: pp., pages, in press, accepted for publication, vol., volume, no., number, et. al, isbn. The following characters are removed -&:()[].

The methods we tested for identifying citations to identical articles are as follows:

1. *Baseline Simple.* A simple baseline method which iterates through all citations. For each citation, we find the maximum number of words which matches with a previous citation, normalized by the length of the shorter citation. If this number exceeds a threshold then the new citation is considered to be a citation to the same article as the previous citation, and the new citation is grouped with the previous citation. Otherwise, a new group is made for the citation.
2. *Word Matching.* An algorithm similar to the Baseline Simple algorithm which first sorts the citations by length, from the longest to the shortest.
3. *Word and Phrase Matching.* The Word Matching algorithm where sequences of two words within each subfield are also considered as terms in the matching process, i.e. this algorithm takes into account some of the ordering of the words, which is ignored by the previous algorithms. Separate thresholds are used for the single word and two word matches.
4. *LikeIt.* A method based on the LikeIt intelligent string comparison algorithm introduced by Yianilos [19, 18]. LikeIt is an sophisticated form of edit distance which tries to build an optimal weighted matching of the letters and multigraphs (groups of letters). LikeIt provides a distance measure between two citations. We use LikeIt in the following way. The citations are ordered by length, from longest to shortest, and we iterate through the citations. For each citation, we find the group with the smallest LikeIt distance, if this distance is below a threshold then the citation is considered identical and added to the previous group, otherwise a new group is created.

In order to evaluate the algorithms, we created three sets of citations taken from the neural network literature (corresponding to queries for the terms “Giles”, “recurrent”, and “fuzzy”). These sets contained 233, 377, and 941 citations respectively. We manually grouped the citations such that each group contained citations to the same paper. We ran the above algorithms on these sets and computed an error score for each case. The error score is the percentage of the groups in the correct grouping for which the automated grouping is not 100% correct. The results are shown in Table 2. The average percentage of the correct groups which were not 100% correct in the automated grouping were, from best to worst: Word and Phrase Matching 7.7%, Word Matching 10.0%, Baseline Simple 10.7%, LikeIt 16.7%. Note that an error of 10%, for example, does not imply that 10% of citations are not grouped correctly, e.g. the addition of one incorrect citation to a group marks the entire group as incorrect.

	Giles	Recurrent	Fuzzy	Average
Number of citations	233	377	941	
Baseline Simple	12%	9%	11%	10.7%
Word Matching	13%	8%	9%	10.0%
Word & Phrase Matching	10%	4%	9%	7.7%
LikeIt	22%	11%	17%	16.7%

Table 2: Results for grouping identical citations with normalization. Results are the percentage of groups in the correct grouping for which the automated grouping is not 100% correct.

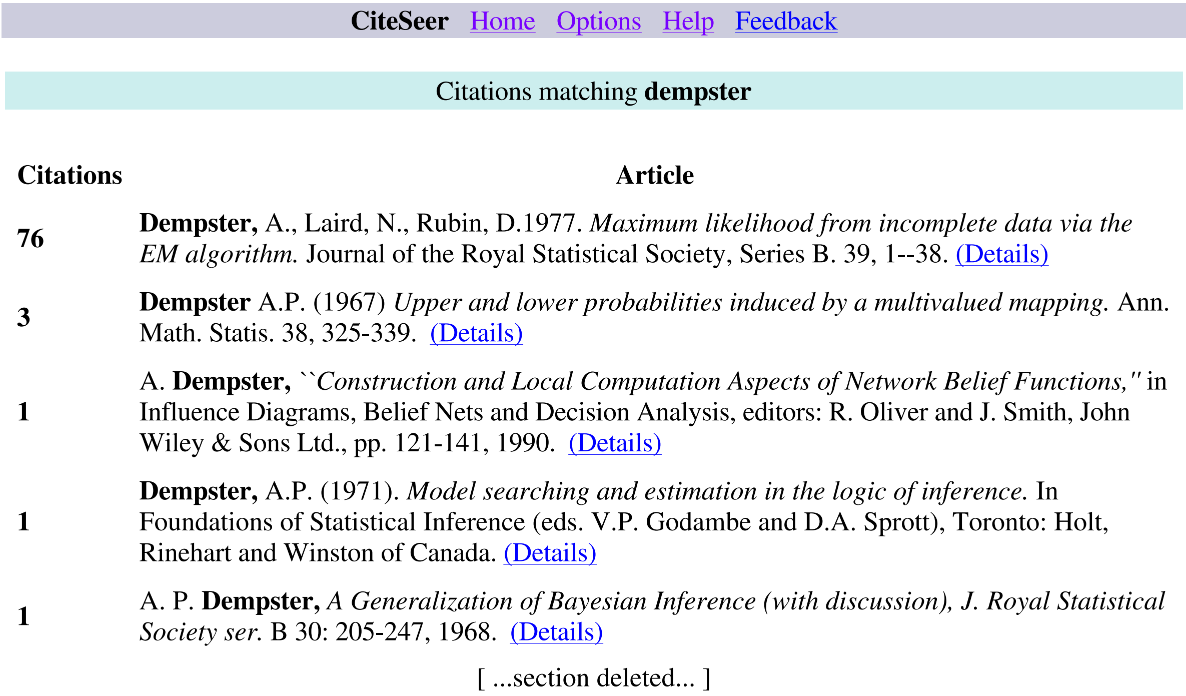
LikeIt currently performs poorly. One explanation may be that two citations to the same article can be of significantly different lengths. LikeIt does not currently support a containment operation (“is a contained in b?” rather than “is a equal to b?”) although there are plans to add such an operation. Alternatively, string distances may not be a good paradigm for determining the semantic similarity of citations, or we may not be using the LikeIt distance in the best manner. LikeIt considers citation strings at the level of letters instead of words, and the order of letters (and thus words). Both of these factors contribute to string differences which may not correspond to semantic differences.

Query and Browsing

The first query to CiteSeer is a keyword search, which can be used to return a list of citations matching the query, or a list of indexed articles. The literature can then be browsed by following the links between the articles made by citations. Figure 1 shows a sample response for the query “dempster” in a CiteSeer database of neural network literature. A list of citations matching the query are shown, ranked by the number of citations to them. Once an initial keyword search is made, the user can browse the database using citation links. The user can find which papers are cited by a particular publication and which papers cite a particular publication, including the context of those citations. Figure 2 lists the papers which cite the first article in Figure 1, along with the context of the citations (obtained by clicking on the appropriate (Details) link in figure 1). An example of full text search in the indexed articles is shown in Figure 3. Here the header information is given for documents which contain the keywords `conjugate and gradient`. Details of a particular document can be found by choosing the link `(Details)`. The details for a sample document is shown in Figure 4. The header, abstract, and list of references made by this document can be seen.

FINDING RELATED DOCUMENTS

Given a database of documents, a user may find a document of interest and then want to find other, related documents. He or she may do this manually by using semantic features such



CiteSeer Home Options Help Feedback

Citations matching **dempster**

Citations	Article
76	**Dempster,** A., Laird, N., Rubin, D.1977. *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the Royal Statistical Society, Series B. 39, 1--38. (Details)
3	**Dempster** A.P. (1967) *Upper and lower probabilities induced by a multivalued mapping.* Ann. Math. Statis. 38, 325-339. (Details)
1	A. **Dempster,** ``*Construction and Local Computation Aspects of Network Belief Functions,"* in Influence Diagrams, Belief Nets and Decision Analysis, editors: R. Oliver and J. Smith, John Wiley & Sons Ltd., pp. 121-141, 1990. (Details)
1	**Dempster,** A.P. (1971). *Model searching and estimation in the logic of inference.* In Foundations of Statistical Inference (eds. V.P. Godambe and D.A. Sprott), Toronto: Holt, Rinehart and Winston of Canada. (Details)
1	A. P. **Dempster,** *A Generalization of Bayesian Inference (with discussion), J. Royal Statistical Society ser.* B 30: 205-247, 1968. (Details)

[...section deleted...]

Figure 1: CiteSeer response for the query "dempster" in a sample database of neural network literature.

as author, research group, or publication venue for the document. However, CiteSeer can also find related documents using various measure of similarity.

There has been much interest in computing the distance (or the inverse, similarity) between a pair of documents (text strings). Most of the known distance measures between bodies of text rely on models of similarity of groups of letters in the text. One type of text distance measure is the *string distance* or *edit distance* which considers distance as the amount of difference between strings of symbols. For example, the *Levenshtein distance* [11] is a well known early edit distance where the difference between two text strings is simply the number of insertions, deletions, or substitutions of letters required to transform one string into another. A more recent and sophisticated example is LikeIt, as mentioned earlier.

Another type of text string distance is based on statistics of words which are common to sets of documents, especially as part of a corpus of a large number of documents. One commonly used form of this measure, based on word frequencies, is known as *term frequency* × *inverse document frequency* (TFIDF) [16]. Sometimes only the stems of words are considered instead of complete words. An often used stemming heuristic introduced by Porter [13] tries to return the same stem from several forms of the same word (e.g. "walking", "walk", "walked" all become simply "walk"). In a document d, the frequency of each word stem s is f_{ds} and the number of documents having stem s is n_s. In document d the highest term frequency is called $f_{d_{max}}$. In one such TFIDF scheme [15] a word weight w_{ds} is calculated as:

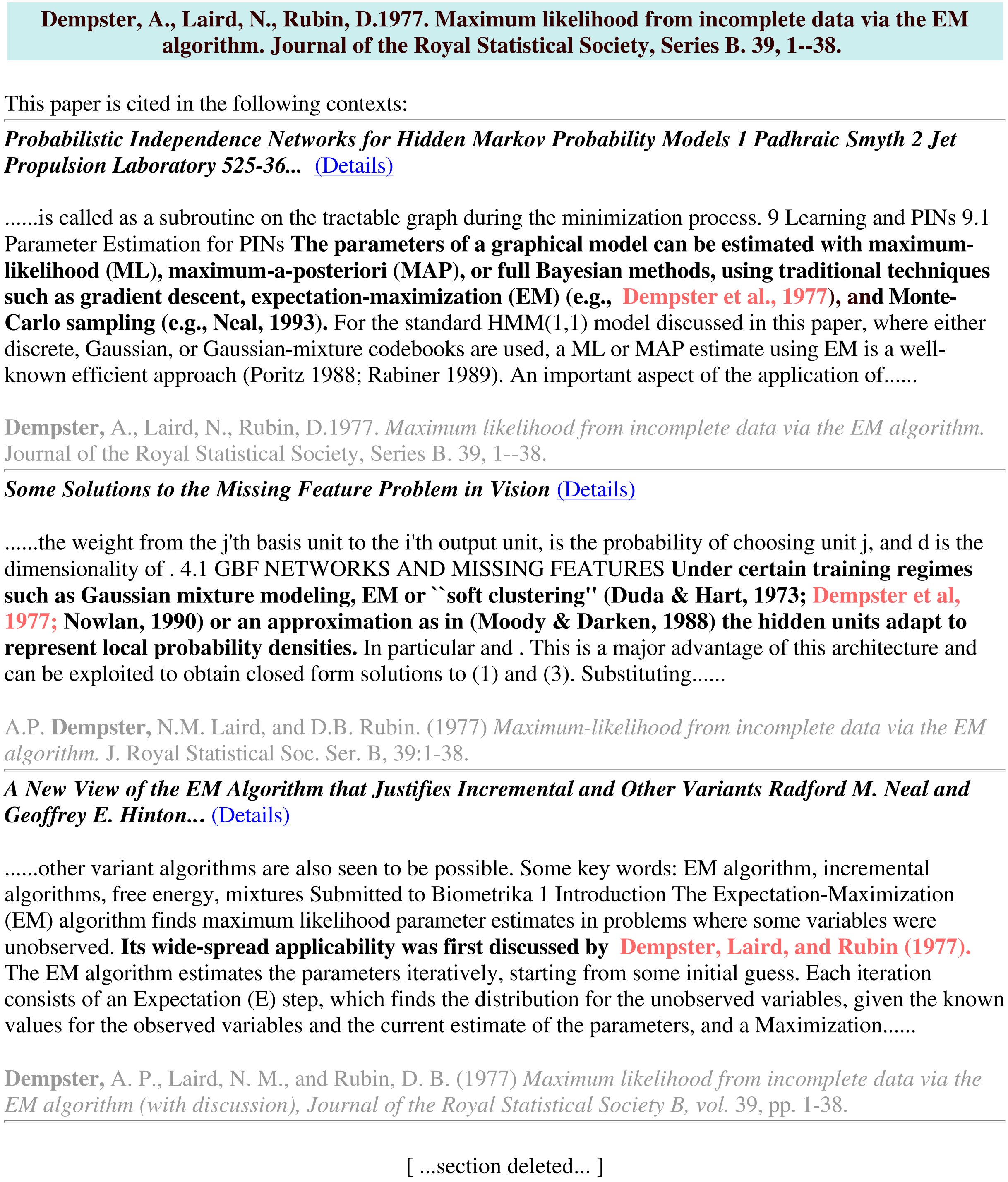
$$w_{ds} = \frac{(0.5 + 0.5\frac{f_{ds}}{f_{d_{max}}})(\log\frac{N_D}{n_s})}{\sqrt{\sum_{j\in d}((0.5 + 0.5\frac{f_{dj}}{f_{d_{max}}})^2(\log\frac{N_D}{n_j})^2)}} \qquad (1)$$

where N_D is the total number of documents. In order to find the distance between two documents, a dot product of the two word vectors for those documents is calculated. One limitation of this approach is the inherent noise – uncommon words may be shared by documents by coincidence, thereby giving false evidence that the documents are related. Another limitation of this approach is the ambiguity of words and phrases. For example "arm" could mean a human limb, or a weapon. Simple word frequencies do not differentiate these uses, requiring context analysis for separation.

A third type of semantic distance measure is one in which knowledge about document components or structure is used. In the case of research publications for example, the citation information can be used for computing similarity.

CiteSeer uses these three methods for computing similarity:

Word Vectors We have implemented a TFIDF scheme to measure a value of each word stem in each document where a vector of all of the word stem values represent the "location" of a document in a word vector space. The projection of the word vector of one document on another document (dot product of the vectors) is the distance measure used. Currently, we use only the top 20 components of each document for computational reasons, however there is evidence



Dempster, A., Laird, N., Rubin, D.1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B. 39, 1--38.

This paper is cited in the following contexts:

Probabilistic Independence Networks for Hidden Markov Probability Models 1 Padhraic Smyth 2 Jet Propulsion Laboratory 525-36... (Details)

......is called as a subroutine on the tractable graph during the minimization process. 9 Learning and PINs 9.1 Parameter Estimation for PINs **The parameters of a graphical model can be estimated with maximum-likelihood (ML), maximum-a-posteriori (MAP), or full Bayesian methods, using traditional techniques such as gradient descent, expectation-maximization (EM) (e.g., Dempster et al., 1977), and Monte-Carlo sampling (e.g., Neal, 1993).** For the standard HMM(1,1) model discussed in this paper, where either discrete, Gaussian, or Gaussian-mixture codebooks are used, a ML or MAP estimate using EM is a well-known efficient approach (Poritz 1988; Rabiner 1989). An important aspect of the application of......

Dempster, A., Laird, N., Rubin, D.1977. *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the Royal Statistical Society, Series B. 39, 1--38.

Some Solutions to the Missing Feature Problem in Vision (Details)

......the weight from the j'th basis unit to the i'th output unit, is the probability of choosing unit j, and d is the dimensionality of . 4.1 GBF NETWORKS AND MISSING FEATURES **Under certain training regimes such as Gaussian mixture modeling, EM or ``soft clustering'' (Duda & Hart, 1973; Dempster et al, 1977; Nowlan, 1990) or an approximation as in (Moody & Darken, 1988) the hidden units adapt to represent local probability densities.** In particular and . This is a major advantage of this architecture and can be exploited to obtain closed form solutions to (1) and (3). Substituting......

A.P. **Dempster,** N.M. Laird, and D.B. Rubin. (1977) *Maximum-likelihood from incomplete data via the EM algorithm.* J. Royal Statistical Soc. Ser. B, 39:1-38.

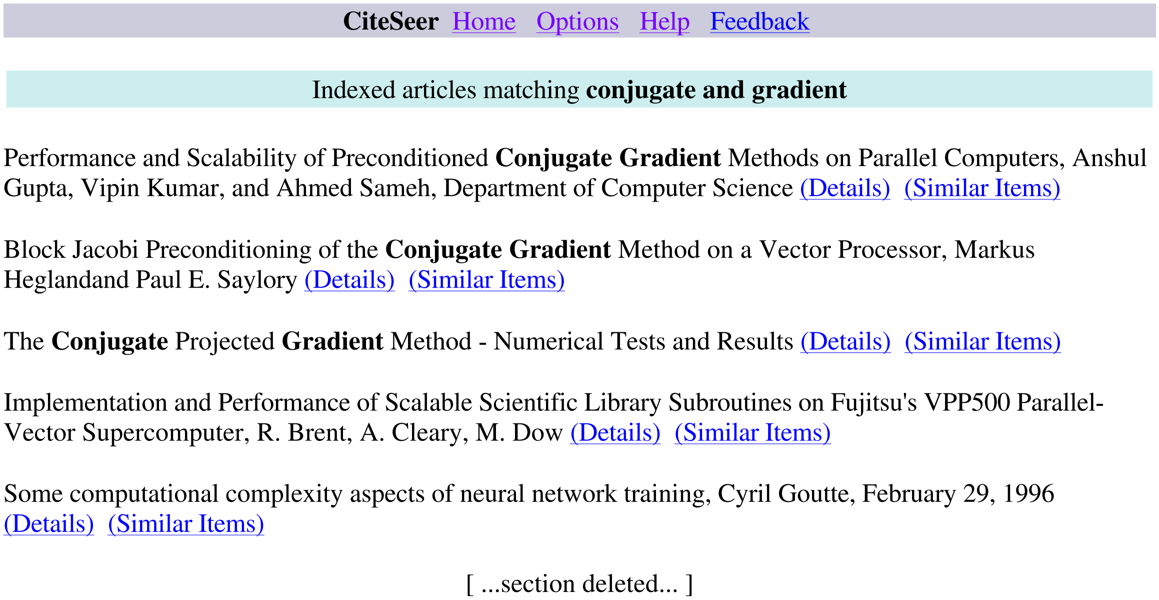
A New View of the EM Algorithm that Justifies Incremental and Other Variants Radford M. Neal and Geoffrey E. Hinton... (Details)

......other variant algorithms are also seen to be possible. Some key words: EM algorithm, incremental algorithms, free energy, mixtures Submitted to Biometrika 1 Introduction The Expectation-Maximization (EM) algorithm finds maximum likelihood parameter estimates in problems where some variables were unobserved. **Its wide-spread applicability was first discussed by Dempster, Laird, and Rubin (1977).** The EM algorithm estimates the parameters iteratively, starting from some initial guess. Each iteration consists of an Expectation (E) step, which finds the distribution for the unobserved variables, given the known values for the observed variables and the current estimate of the parameters, and a Maximization......

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977) *Maximum likelihood from incomplete data via the EM algorithm (with discussion), Journal of the Royal Statistical Society B, vol.* 39, pp. 1-38.

[...section deleted...]

Figure 2: CiteSeer response showing the context of citations to the article “Maximum likelihood from incomplete data via the EM algorithm”.



CiteSeer Home Options Help Feedback

Indexed articles matching **conjugate and gradient**

Performance and Scalability of Preconditioned **Conjugate Gradient** Methods on Parallel Computers, Anshul Gupta, Vipin Kumar, and Ahmed Sameh, Department of Computer Science (Details) (Similar Items)

Block Jacobi Preconditioning of the **Conjugate Gradient** Method on a Vector Processor, Markus Heglandand Paul E. Saylory (Details) (Similar Items)

The **Conjugate** Projected **Gradient** Method - Numerical Tests and Results (Details) (Similar Items)

Implementation and Performance of Scalable Scientific Library Subroutines on Fujitsu's VPP500 Parallel-Vector Supercomputer, R. Brent, A. Cleary, M. Dow (Details) (Similar Items)

Some computational complexity aspects of neural network training, Cyril Goutte, February 29, 1996 (Details) (Similar Items)

[...section deleted...]

Figure 3: CiteSeer response for documents containing the words conjugate and gradient.

that this truncation may not have a large effect on the distance measures [15].

String Distance CiteSeer uses the LikeIt string distance [18] to measure the edit distance between the headers of documents in a database. The header of a document is simply all of the text in a document before the abstract (or the introduction if there is no abstract). The header tends to contain items such as the document title, author name and affiliation, and possibly the publication venue. LikeIt tries to match substrings in a larger string – common authors, institutions, or words in the title will tend to reduce the LikeIt distance between headers. The premise behind the use of LikeIt is that the document header contains very important information about the document, and that the presence of words in similar arrangements indicates documents of similar origin.

Citations Single words (and even phrases to a lesser degree) may not have much power to represent the topic of or concepts discussed in a research paper. Citations of other works on the other hand, are hand picked by the authors as being related documents. It seems intuitive then to use citation information to judge the relatedness of documents. CiteSeer uses common citations to make an estimate of which documents in the downloaded database of research papers are the most closely related to a document picked by the user. This measure, “Common Citation × Inverse Document Frequency” (CCIDF) is analogous to the word oriented TFIDF [15] word weights. The algorithm to calculate the CCIDF relatedness of all documents in the database to a document of interest A and choose the best M documents is:

1. Assign a weight (w_i) to each citation i, equal to the inverse of the frequency of the citation in the entire database.
2. Determine the list of citations and their associated weights for document A and query the database to find the set of n documents $\{B_j\} : j = 1 \dots n$ which share at least one citation with A.
3. For each $j = 1 \dots n$, determine the relatedness of the document R_j as the sum of the weights of the citations shared with A.

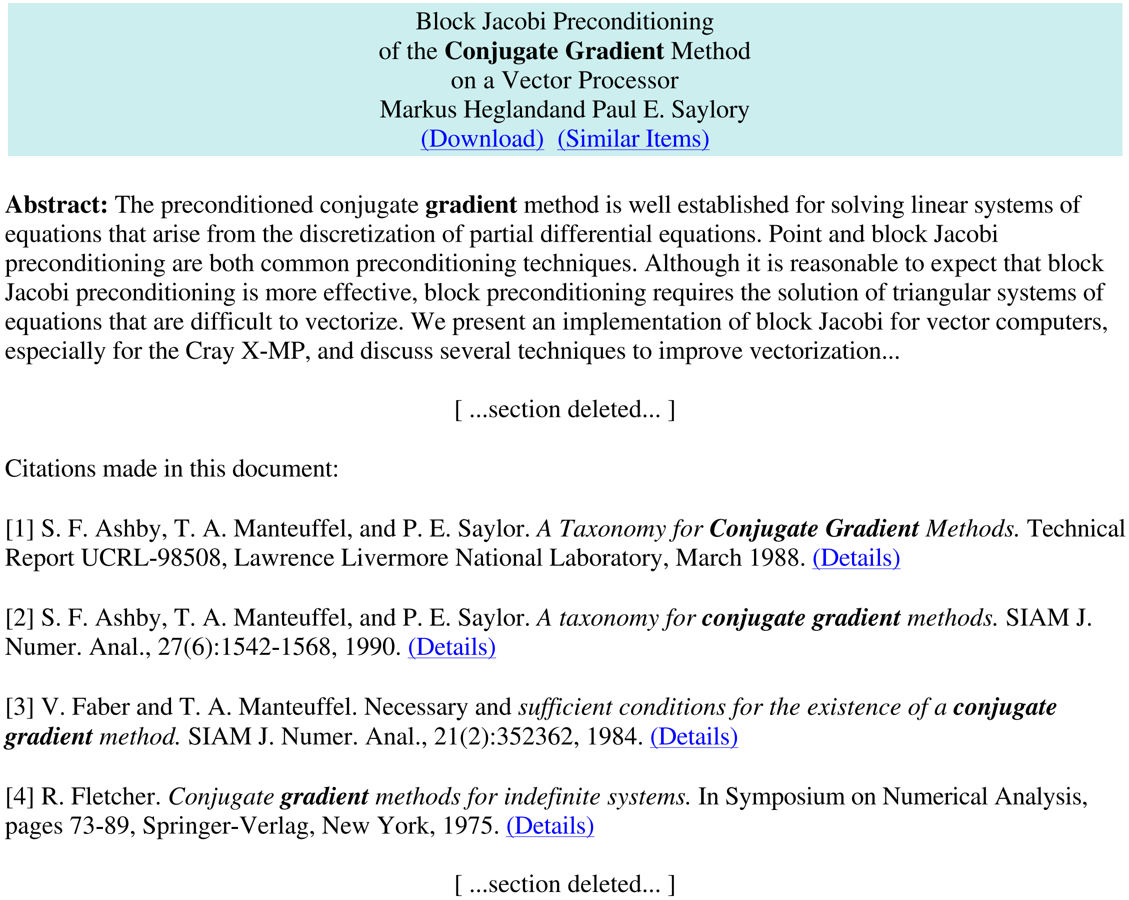
$$R_j = \sum_{i \in A \cap i \in B_j} w_i \tag{2}$$

4. Sort the R_j values and return the documents B_j with the M highest R_j values.

As in the use of TFIDF, CCIDF assumes that if a very uncommon citation is shared by two documents, this should be weighted more highly than a citation made by a large number of documents. Although we have not formally measured the performance of CCIDF, we have found it to be useful, and to perform better than the word vector or LikeIt based automatic similar document retrieval methods.

Combination of Methods Although we have found that citation based similar document retrieval is subjectively better than word vector or LikeIt based retrieval, we plan to more thoroughly evaluate the performance of CCIDF and compare with other techniques, such as the word vector and edit distance methods presented here.

In order to use a similarity distance measure that may be more accurate than any single method alone, CiteSeer combines the different methods of document similarity



Block Jacobi Preconditioning
of the **Conjugate Gradient** Method
on a Vector Processor
Markus Heglandand Paul E. Saylory
(Download) (Similar Items)

Abstract: The preconditioned conjugate **gradient** method is well established for solving linear systems of equations that arise from the discretization of partial differential equations. Point and block Jacobi preconditioning are both common preconditioning techniques. Although it is reasonable to expect that block Jacobi preconditioning is more effective, block preconditioning requires the solution of triangular systems of equations that are difficult to vectorize. We present an implementation of block Jacobi for vector computers, especially for the Cray X-MP, and discuss several techniques to improve vectorization...

[...section deleted...]

Citations made in this document:

[1] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor. *A Taxonomy for **Conjugate Gradient** Methods.* Technical Report UCRL-98508, Lawrence Livermore National Laboratory, March 1988. (Details)

[2] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor. *A taxonomy for **conjugate gradient** methods.* SIAM J. Numer. Anal., 27(6):1542-1568, 1990. (Details)

[3] V. Faber and T. A. Manteuffel. Necessary and *sufficient conditions for the existence of a **conjugate gradient** method.* SIAM J. Numer. Anal., 21(2):352362, 1984. (Details)

[4] R. Fletcher. *Conjugate **gradient** methods for indefinite systems.* In Symposium on Numerical Analysis, pages 73-89, Springer-Verlag, New York, 1975. (Details)

[...section deleted...]

Figure 4: An example of detailed document information.

mentioned above. The combined similarity measure is a weighted sum of the individual similarity measures as described in the following algorithm:

1. Calculate the word vector, LikeIt, and citation similarity measures and normalize each measure to a 0 to 1 scale where 1 represents semantically identical documents, and 0 represents completely different documents (infinite semantic distance). Label the normalized similarity measures between two documents A and B as $WV(A,B)$, $LI(A,B)$, and $CI(A,B)$ respectively.
2. Given a target document A and a set of n candidate documents $\{B_j\} : j = 1 \ldots n$, measure the similarity between A and all n of the B_j documents using the three measures from step 1.
3. Let w_{WV}, w_{LI}, w_{CI} be the weights given to their respective similarity measures. These weight values are between 0 and 1 and they are normalized so that $w_{WV} + w_{LI} + w_{CI} = 1$.
4. Find a combined similarity measure S_j between A and each of the B_j documents as the weighted sum:
$$\begin{aligned} S_j &= w_{WV}WV(A,B) + w_{LI}LI(A,B) \\ &\quad + w_{CI}CI(A,B) \end{aligned} \tag{3}$$
5. Retrieve the documents with the highest S_j values.

Currently, we have manually set the weights to the fixed values $WV(A,B) = 0.25$, $LI(A,B) = 0.25$, and $CI(A,B) = 0.50$. In the future, we intend to explore the use of learning techniques to automatically determine the best weights. As an example, Figure 5 shows the documents CiteSeer found in an image compression database to be most similar to the paper, "On the Improvements of Embedded Zerotree Wavelet (EZW) Coding".

FUTURE WORK

There are a number of areas for future work on CiteSeer. For example: a) the related/similar document retrieval system could be enhanced and improved. We plan to explore new distance measures, refine our existing distance measures, continue to investigate combining different types of distance measures, and explore the use of learning techniques. We plan to investigate learning user notions of interesting research topics from both user activity and user feedback. Also, in the CCIDF dot product distance between documents, we intend to include the use of more specific citation information to judge distance between documents. For example, in addition to common papers, citation of the same authors or journals may indicate a relationship. b) the

Query: `wavelet`

On the Improvements of Embedded Zerotree Wavelet (EZW) Coding, Jin Li and Po-Yuen Cheng and C.-C. Jay Kuo, Signal and Image Processing Institute and Department of Electrical Enineering-Systems, University of Southern California, Los Angeles, California 90089-2564

Related papers:

Similarity = **0.525**

IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, June 1996
A New Fast and Efficient Image Codec Based
on Set Partitioning in Hierarchical Trees \Lambda
Amir Said
Faculty of Electrical Engineering, P.O. Box 6101
State University of Campinas (UNICAMP), Campinas, SP, 13081, Brazil
William A. Pearlman
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY, 12180, U.S.A.
(Details) (Download) (Find Similar Items)

Abstract: Embedded zerotree wavelet (EZW) coding, introduced by J. M. Shapiro, is a very effective and computationally simple technique for image compression. Here we offer an alternative explanation of the principles of its operation, so that the reasons for its excellent performance can be better understood. These principles are partial ordering by magnitude with a set partitioning sorting algorithm, ordered bit plane transmission, and exploitation of self-similarity across different scales of an image wavelet transform. Moreover, we present a new and different implementation, based on set partitioning in hierarchical trees (SPIHT), which provides even better performance than our previosly reported extension of the EZW that surpassed the performance of the original EZW. The image coding results, calculated from actual file sizes and images reconstructed by the decoding algorithm, are either compara- ble to or surpass previous results obtained through much more sophisticated and computationally complex methods. In addition, the new coding and decoding pro- cedures are extremely fast, and they can be made even faster, with only small loss in performance, by omitting entropy coding of the bit stream by arithmetic code. I.

Similarity = **0.46785714285714**

SPIE's 1996 Symp. on Visual Communications and Image Processing '96
Three-Dimensional Subband Coding of Video
Using the Zero-Tree Method
Yingwei Chen and William A. Pearlman
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180
yingwei@ipl.rpi.edu, pearlman@ecse.rpi.edu
(Details) (Download) (Find Similar Items)

Abstract: In this paper, a simple yet highly effective video compression technique is presented. The zero-tree method by Said [3,4], which is an improved version of Shapiro's [1,2] original one, is applied and expanded to three-dimension to encode image sequences. A three-dimensional subband transfor- mation on the image sequences is first performed, and the transformed information is then encoded using the zero-tree coding scheme. The algorithm achieves results comparable to MPEG-2, with- out the complexity of motion compensation. The reconstructed image sequences have no blocking effects at very low rates, and the transmission is progressive. Key word list: video compression, subband coding, video coding, progressive transmission, image sequence coding 1

[...section deleted...]

Figure 5: CiteSeer response for documents similar to one of interest in an image compression manuscript database.

heuristics used to locate articles could be improved, e.g. the modification frequency of pages could be used to set an interval which is used to check for updates to a page and relevant newsgroups and mailing lists could be monitored in order to index new papers as soon as they are announced. c) the database could be augmented in order to maintain user feedback on each article, e.g. comments or questions regarding the article, followup comments or responses from the author or others, or ranking information. d) CiteSeer could easily be extended in order to keep track of topics and articles that users are interested in. Users could then be automatically informed when new articles are indexed which match their interests. For example: profiles could be kept for topics which a user is interested in. The profiles could be updated autonomously by watching user actions (e.g. the downloading of a particular document), or they could be updated with relevance feedback. Alternatively, users may register papers of interest so that they can be informed of new articles which cite these papers.

SUMMARY

CiteSeer is an autonomous citation indexing system. CiteSeer is different from previous citation indexing systems in that the indexing process is completely automatic – CiteSeer autonomously locates, parses, and indexes articles found on the World Wide Web. The advantages of CiteSeer include timeliness, automation, and browsing of citation context. The system indexes preprints and technical reports, as well as journal and conference articles. The publication delay for journals and conferences means that CiteSeer has access to more recent articles. This allows scientific effort to progress more rapidly, and reduces the possibility of performing work which has already been done by others. The system is completely automatic, and does not require human effort in the indexing process.

Due to the organization of the Web, CiteSeer is not currently able to provide as comprehensive an index as the traditional systems because many publications are not currently available on-line. This is changing rapidly however, and we expect that this disadvantage will disappear over time. Also, because of potential difficulties, for example in the automatic identification and disambiguation of authors, the CiteSeer index may not be as accurate as traditional manual indexes. However, CiteSeer may allow a human to assess the accuracy of results (by browsing detailed information and citation context in the database) more easily than with traditional citation indexing systems.

Some of the most important capabilities of CiteSeer are: a) the ability to parse citations from papers and identify citations to identical papers that may differ in syntax. This allows the generation of citation statistics and the grouping of work which cites a given paper, b) the ability to extract and show the context of citations to a given paper, allowing a researcher to see what others authors have to say about a given paper, and c) the ability to find related articles based on common citations and on word vector or string distance similarity.

Acknowledgments

We would like to thank Peter Yianilos for useful discussions and for assistance using LikeIt. We would also like to thank Bob Krovetz and Majd Sakr for helpful discussions and the referees for useful suggestions.

REFERENCES

1. Eytan Adar and Jeremy Hylton. On-the-fly hyperlink creation for page images. In *Proceedings of Digital Libraries '95 -The Second Annual Conference on the Theory and Practice of Digital Libraries*, June 1995.
2. T. A. Brooks. Evidence of complex citer motivations. *Journal of the American Society for Information Science*, 37:34–36, 1986.
3. Robert D. Cameron. A universal citation database as a catalyst for reform in scholarly communication. *First Monday*, 2(4), 1997.
4. S. E. Cozzens. What do citations count? The rhetoric-first model. *Scientometrics*, 15:437–447, 1989.
5. Blaise Cronin and Herbert W. Snyder. Comparative citation rankings of authors in monographic and journal literature: a study of sociology. *Journal of Documentation*, 53(3):263–273, 1997.
6. Eugene Garfield. *Citation Indexing: Its Theory and Application in Science, Technology, and Humanities*. Wiley, New York, 1979.
7. Eugene Garfield. The concept of citation indexing: A unique and innovative tool for navigating the research literature. *Current Contents*, January 3, 1994.
8. Eugene Garfield. Where was this paper cited? *Current Contents*, January 31, 1994.
9. G. N. Gilbert. Referencing as persuasian. *Social Studies of Science*, 7:113–122, 1977.
10. `http://www.isinet.com` Institute for Scientific Information, 1997.
11. V. I. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones (original in Russian). *Russian Problemy Peredachi Informatsii*, 1:12–25, 1965.
12. M. H. MacRoberts and B. R. MacRoberts. The negational reference: Or the art of dissembling. *Social Studies of Science*, 14:91–94, 1984.
13. M. F. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, 3 1980.
14. Gerard Salton. Automatic indexing using bibliographic citations. *Journal of Documentation*, 27:98–110, 1971.
15. Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Tech Report 87-881, Department of Computer Science, Cornell University, 1987.
16. Gerard Salton and