CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications

Kurt D. Bollacker[1,2], Steve Lawrence[2], and C. Lee Giles[2,3]
{kurt,lawrence,giles}@research.nj.nec.com

[1] University of Texas at Austin, Austin, TX 78712
[2] NEC Research Institute, Princeton, NJ 08540
[3] UMIACS, University of Maryland, College Park, MD 20742

Abstract

Published research papers available on the World Wide Web (WWW or Web) are often poorly organized, often exist in non-text form (e.g. Postscript) documents, and increase in quantity daily. Significant amounts of time and effort are commonly needed to find interesting and relevant publications on the Web. We have developed a Web based information agent that assists the user in the process of performing a scientific literature search. Given a set of keywords, the agent uses Web search engines and heuristics to locate and download papers. The papers are parsed in order to extract information features such as the abstract and individually identified citations which are placed into an SQL database. The agent's Web interface can be used to find relevant papers in the database using keyword searches, or by navigating the links between papers formed by the citations. Links to both "citing" and "cited" publications can be followed. In addition to simple browsing and keyword searches, the agent can find papers which are similar to a given paper using word information and by analyzing common citations made by the papers.

1 Introduction

Scientific research attempts to add to the body of human knowledge, but because the realm of research is so vast, researchers have the potential to duplicate previously performed work. A literature search for relevant published research results is generally used to avoid duplication of work. Most published scientific research appears in paper documents such as scholarly journals or conference proceedings, but there is usually a considerable time lag between the completion of research and the availability of such publications. The World Wide Web (WWW or Web) has become an important distribution medium for scientific research because Web publications are often available before any corresponding printed publications in journals or conference proceedings. In order to keep up with current research, especially in quickly advancing fields, a researcher can use the Web to download papers as soon as they are made available by the author.

A problem in the search for current relevant published research is the exponential growth of the literature. The Web makes literature easier to access, but ease of publication encourages an increased publication rate. Additionally, Web based research publications tend to be poorly organized (each institution or researcher may have his or her own organizational scheme), and are spread throughout the Web. Despite these problems, there are potentially important advantages to Web based scientific literature – articles on the web can be retrieved and processed by autonomous agents much more easily than printed documents. Agents searching the Web can provide an automated means to find, download, and judge the relevance of published research contained therein.

1.1 Search Engines and Web Browsing

Currently, one of the most commonly used methods for finding interesting publications on the Web is to use a combination of *Web Search Engines* with manual Web browsing. Web search engines such as AltaVista (`http://altavista.digital.com`) index the text contained on Web pages, allowing users to find information using keyword search. Some research publications on the Web are made available in HTML format, making the text of these papers searchable with Web search engines. However, most of the published research papers on the Web are in Postscript form (which preserves the formatting of the original), rather than HTML. The text of these papers is not indexed by search engines such as AltaVista, requiring researchers to locate pages which contain links to these papers (e.g. by searching for a paper title or author name).

1.2 An Agent to Assist in Finding Relevant Publications

The "mostly manual" method of finding literature using search engines and browsing requires a great deal of tedious, repetitive user intervention in order to reach a point where the user can actually read part of the document to determine whether it is of interest. Additionally, even when papers are immediately available, there may be too many potentially interesting papers to practically peruse. In order to assist the user in finding relevant Web based research publications, we have developed *CiteSeer*, an "assistant agent" which improves upon this manual process in three ways:

1. It automates the tedious, repetitive, and slow process of finding and retrieving Web based publications.
2. Once potentially relevant papers are retrieved, it guides the user towards interesting papers by making them searchable.
3. When a relevant paper is found, it helps the user by suggesting other related papers using similarity measures derived from semantic features of the retrieved documents.

The operation of CiteSeer is relatively simple. Given a set of broad topic keywords, CiteSeer uses Web search engines and heuristics

Published in: *2nd International ACM Conference on Autonomous Agents*, pp. 116-123, ACM Press, May, 1998.

to locate and download papers which are potentially relevant to the user's topic. The downloaded papers are parsed to extract semantic features, including citations and word frequency information. This information is then stored in a database which the user can search by keyword, or use citation based links to find related papers. The agent can also automatically find papers similar to a paper of interest using semantic feature information.

2 Previous Research

The design of CiteSeer takes benefit from three broad lines of previous research. One is work in the area of Web, interface, and assistant software agents. Another line of previous research is investigation into semantic distance measures between text documents so that agents can simulate a user's concept of document similarity. One important example of a feature used to form semantic distance measures is that of *citation indexing* which records published research citations of and by other publications.

2.1 Assistant Agents

Assistant Agents are often defined as agents designed to assist the user with the use of software systems by performing tasks on behalf of the user, making interaction with the software system easier and/or more efficient. Several Web based assistant agents have been constructed to help the user find interesting and relevant World Wide Web pages more quickly and easily. Some of these, such as [10, 3, 9, 11] (and [5] contains an overview of several agents) learn from user feedback in an environment of word vector features to find more relevant Web pages. Interesting changes to known relevant Web pages are learned by the "Do-I-Care" agent [17]. This system also allows the agent to learn from the feedback of another user. Although it does no learning, the heuristic Web agent "CiFi" [8] tries to find citations of a specified paper on the World Wide Web. CiteSeer differs from most previous Web agents in that it actually creates a customized "view" of a part of the Web. A local database is created which structures documents downloaded from the web in a way that is far more easily searched and browsed than if a simple list of URLs were presented. Additionally, CiteSeer allows searching inside Postscript documents, which are "opaque" to all previous search engines and agents.

2.2 Semantic Distance Measures

Given a set of documents (essentially text strings), there has been much interest in estimating a human notion of distance (or the inverse, similarity) measurements between documents. Presently, we are aware of three commonly used types of models. One is the *string distance* or *edit distance* measure which considers distance as the amount of difference between strings of symbols. For example, the *Levenshtein distance* [7] 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 to transform one string into another. A more recent and sophisticated example is LikeIt [18, 19] where a string distance is based on an algorithm that tries to "build an optimal weighted matching of the letters and multigraphs (groups of letters)".

Another type of text string distance measure 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) [15]. Consider a dictionary of all of the words (terms) in a corpus of documents. In some systems, very common words, sometimes called *stop words*, such as *the*, *a*, etc. are ignored for computational efficiency. Also, sometimes only the stems of words are considered instead of complete words. An often used stemming heuristic introduced by Porter [12] 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}, the number of documents having stem s is n_s, and the highest term frequency is called $f_{d_{max}}$. In one such TFIDF scheme [14] a word weight w_{ds} is calculated as:

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

where N_D is the total number of documents. In order to find the distance between two documents, a simple dot product of the two word vectors for those documents is calculated.

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, citations of papers by other papers has been used to create *citation indices* which can be used to gauge document relatedness [13]. Another example is the ParaSite system [16], in which the nearness of links to referenced Web pages in the HTML structure of a referencing Web page are used as an indicator of relatedness of the referenced pages.

2.3 Citation Indexing

References contained in scientific articles are used to give credit to previous work in the literature and can be thought of as a link between the "citing" and "cited" articles. A citation index contains the references that an article cites, linking the articles with the cited works. Citations are a *semantic feature* of a research publication which can be used to determine its relationships to other publications. Citation indices were originally designed mainly for information retrieval [6]. 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?) making it a powerful tool for literature search.

There are a few existing commercial citation indexed databases, such as those provided by the Institute for Scientific Information (ISI) [1]. ISI produces several citation indices including the *Science Citation Index*®, which is a multidisciplinary citation index for scientific periodicals. Another commercial database which provides citation indexing is the legal database offered by the West Group [2], which indexes case law, as opposed to scientific research publications. CiteSeer-created indices are a departure from commercial indices of scientific literature due to their automatic creation and autonomous extraction of citations, and the ability for users to create by users in real time. All previous commercial indices are large, accumulative databases while CiteSeer is an up to date "snapshot" of relevant literature on the web.

2.4 A Universal Citation Database

Cameron has proposed a "universal, [Internet-based,] bibliographic and citation database linking every scholarly work ever written" [4]. He describes a system in which all of the worlds published research would be available to and searchable by any scholar with Internet access. Such a database would be highly "comprehensive and up-to-date", making it a powerful tool for academic literature research. CiteSeer can be thought of as a partial agent implementation of what Cameron would call a "semi-universal citation database", since a CiteSeer agent only gathers works beyond a point in time. Perhaps one of the most important differences between Cameron's 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 document and citation database from downloaded documents, whereas Cameron has proposed a system whereby authors or institutions must make citations in a specific format.

3 Agent Architecture

The CiteSeer agent consists of three main components: (i) a sub-agent to automatically locate and acquire research publications, (ii) a document parser and database creator, and (iii) a database browser interface which supports search by keyword and browsing by citation links. Figure 1 gives a diagram of this architecture.

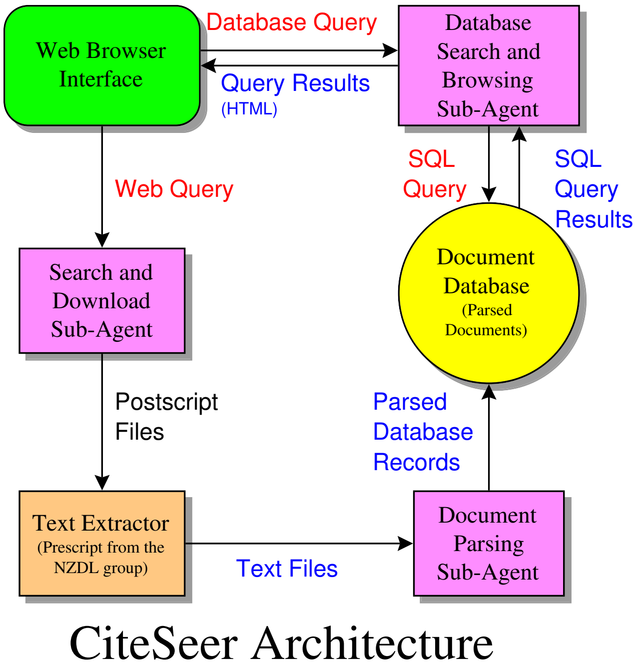


Figure 1: CiteSeer Agent Architecture.

3.1 Document Acquisition

User operation of CiteSeer is relatively straightforward. When the user wishes to explore a new topic, a new instance of the agent is created for that particular topic. The first step is the invocation of a sub-agent to search for Web pages which are likely to contain links to research papers of interest. The user invokes this sub-agent by giving it broad keywords. The agent uses Web search engines (e.g. AltaVista, HotBot, Excite) and heuristics (e.g. searching for pages which also contain the words "publications", "postscript", etc.). The agent locates and downloads Postscript files identified by ".ps", ".ps.Z", or ".ps.gz" extensions. Duplicate URLs and Postscript files are avoided. Although the only supported format of Web based document is Postscript, the vast majority of Web based publications are in this form, making this a minor limitation. Other formats could be used in the future with the appropriate converters.

3.2 Document Parsing

Document parsing is the processing of downloaded documents to extract semantic features from the documents. An instance of the CiteSeer agent invokes a parsing sub-agent to control the various parsing programs, and perform organizational housekeeping, error log generation, and hardware usage management. The parsing programs extract the desired document features and place them into an SQL database. The database contains the following tables:

- *document:* Contains pieces of text from the document, the URL of the document, and a Unique Article ID number (UAID).
- *documentwords:* Contains word frequency information about the body of documents referenced in the *document* table.
- *citation:* Contains the text of citations made by the documents in the *document* table as well as parsed field information. Each record in this table has a Unique Citation ID Number (UCID) and a field for the corresponding UAID.
- *citationwords:* Contains word frequency about the citations in *citation*.
- *citecluster and clusterweights:* Contains cluster number and weighting information when grouping identical citations in different forms. This information is used for automatic similar document retrieval.

As documents are searched for by the parsing sub-agent, a document parsing sub-agent watches the download directory and begins the parsing process on documents as they become available. The first step in document parsing is the extraction of the raw text from the Postscript file. Currently, we use the *pstotext* program from the DEC Virtual Paper Project. This program tries to extract ASCII text formatted using information from the original Postscript text formatting. Then, the formatted ASCII text is verified as a valid research document including a check for the existence of a list of references near the end of the document and a check for non-English documents (Publications in other languages are not yet handled). An attempt is also made to correct the page order of reverse page order documents while invalid documents are recorded as such and skipped. Heuristics are used to identify the following in valid documents:

- *Header:* This is the information at the beginning of the paper that contains the title, author, institution, and other information that comes before actual document text. Identification of features inside the header (e.g. author, title) is not performed as yet.
- *Abstract:* If it exists, the abstract text is extracted.
- *Introduction:* If it exists, the first 300 words of the introduction section are extracted.
- *Citations:* The list of references made by the document are extracted and parsed further as described below.
- *Word Frequency:* Word frequencies are recorded for all words in the document except those in the citations and stop words. The recorded words are stemmed using Porter's algorithm.

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 citation in the body of the document (e.g. "[6]", "[Giles97]", "Marr 1982"). Word frequency of each citation is also recorded, with stop word removal and stemming applied the same as in the document word frequency extraction. Additionally, we use the citation tags to find the locations in the document body text where the citations are actually made. This allows us to extract the context of the citations during database browsing.

The heuristics used to parse citations were constructed with 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 year of publication exists in almost every citation as a four digit number beginning with the digits "19". 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 guess where the desired subfield existed (if at all). For example, author information almost always precedes title information, and publisher almost always follows the title.

3.3 Database Browsing

The third component of the CiteSeer agent is the document database browser. This consists of a query processing sub-agent which takes a user query of proper syntax and returns an HTML formatted response. Typically, the query program is not used directly, but through a Web browser interface. The query processing sub-agent provides several different browsing capabilities that allow a user to easily navigate through the document database. Although search by keyword is supported, there is emphasis on using the links between "citing" and "cited" documents to find related research papers.

The first access to the publication database must be a keyword search. After any non-empty query response is given, then the user may browse. A CiteSeer database was created using the initial keywords "neural networks" for demonstration purposes. Note that we have not attempted to index all neural network publications on the Web. Suppose the user would like to find all cited papers jointly authored by Giles and Chen therein. The example query, **citation: +Giles +Chen** asks for all citations which contain the words "Giles" and "Chen". Figure 2 shows the results of this query in the sample neural network database. The number of documents which cite each reference is given in brackets before the reference. At the bottom, we can see that there are a total of 36 references to papers by these two authors in the neural network database. We use an identical citation grouping (ICG) algorithm to group several instances of the same cited document which may appear in different formats in the citing documents, as described below.

The first page of results from an example keyword search in the documents themselves, **document: +recurrent +series**, is shown in Figure 3. Here the header information is given for documents which contain the keywords in their body. Details of a particular document can be found by choosing the link `(Details)`. The first page of details of the second item in Figure 3 are shown in Figure 4. The header, abstract, URL, and list of references made by this document can be seen. Once an initial keyword search is made, the user can browse the database by using citation-document 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. Returning to the example of papers authored by Giles and Chen, suppose a user wishes to know which papers cite the article "Extracting and learning an unknown grammar with recurrent neural networks", shown as the third item in Figure 2. There are 9 references to this work in the sample neural network database. Choosing the link `(Details)` following this reference sends a query to CiteSeer's query processor, which returns results (the first page of which is) shown in Figure 5. The user is given the exact form of each citation, a link and URL to the citing document, and the context of the citation in the citing document. If desired, the user can retrieve the details of a citing document by choosing the link to a citing document. The results of such a query are in the same format as Figure 4.

4 Semantic Distance Measures

As mentioned in the references to previous work, semantic distance measures between bodies of text are used to measure their "relatedness". We have implemented semantic distance measures in two applications in CiteSeer. First, we have used word frequency and edit distances to group the different forms of the same citation. Second we have developed a means of using citation frequency information to find documents related to one of a user's interest in the CiteSeer database.

4.1 Identical Citation Grouping

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.
```

As suggested by the example citations above, the problem is not completely trivial, and so we have implemented an identical citation grouping (ICG) method. The first step in this method is a normalization of citations by rules such as conversion to lower case and removal of most punctuation. Then, we use the following word/phrase matching algorithm to group the citations:

• *Sort the citations by length, from the longest to the shortest citation.*
• *For each citation c:*

1. *Find the group g with the highest number of matching words*
2. *Let a = the ratio of the number of non-matching words to the number of matching words*
3. *Let b = the ratio of the number of non-matching phrases to the number of matching phrases where a phrase is every set of two successive words in every section of the citation containing three or more words.*
4. *If (a $<$ threshold1) or (a $<$ threshold2 and b $<$ threshold3) then then add c to the group g, else create a new group for this citation*

End for

In this algorithm, if a citation under consideration is close enough to an existing citation group, then it is included. Otherwise it starts a new group. We have performed a formal quantitative evaluation of this and comparative algorithms (a simple baseline method, a method based on LikeIt, and the above method without phrases), and found that this algorithm performs better than the others (we have not included details of the comparison due to space requirements).

4.2 Finding Similar Documents

Given a database of documents, a user may find a document of interest and then want to find other, related documents. He/she may do this manually by using semantic features such as author, research group, or publication venue for the document. However, CiteSeer also has a mechanism for the automatic retrieval of related documents based on distance measures of semantic features extracted from those documents.

CiteSeer Home Help Suggestions

Query: `citation: +Giles +Chen`

Citations	Article
12	**Giles**, C.L., Sun, G.Z., Chen, H.H., Lee, Y.C., **Chen**, D., (1990) ``*Higher Order Recurrent Networks and Grammatical Inference,"* Advances in Neural Information Processing Systems 2, D.S. Touretzky (ed), Morgan Kaufmann, San Mateo, CA, (1990), p. 380. (Details)
10	**Giles**, C.L., Miller, C.B., **Chen**, D., **Chen**, H.H., Sun, G.Z., & Lee, Y.C. (1992). *Learning and extracting finite state automata with second-order recurrent networks.* Neural Computation, 2, 331-349. (Details)
9	C. **Giles**, C. Miller, D. **Chen,** G. Sun, H. **Chen**, and Y. Lee, ``*Extracting and learning an unknown grammar with recurrent neural networks,"* in Advances in Neural Information Processing Systems 4 (J. Moody, S. Hanson, and R. Lippmann, eds.), San Mateo (Details)
2	T. Maxwell, C. L. **Giles**, Y. C. Lee, and H. H. **Chen**. *Transformation invariance using high order correlations in neural net architectures.* In Proceedings of the IEEE international conference on systems, man, and cybernetics, pages 627--632, October (Details)
2	M. Goudreau, C. **Giles**, **S.** Chakradhar, and D. **Chen**, ``*First-order vs. second-order single layer recurrent neural networks,"* IEEE Transactions on Neural Networks, vol. 5, no. 3, pp. 511--513, 1994. (Details)
1	Y. C. Lee, G. Doolen, H. **Chen**, G. Sun, T. Maxwell, H. Lee, and C. L. **Giles**. *Machine Learning Using a Higher Order Correlation Network.* Physica D: Nonlinear Phenomena, vol. 22, pp. 276--306, 1986. ISSN: 0167-2789 (Details)

36 citations found

Figure 2: Results of a keyword search on citations in the neural network database.

4.2.1 Comparison with Previous Research

Previous Web assistant agents (e.g. [10, 3, 17]) have used word frequency information to automatically measure how related two documents are. While this has been useful in some domains, uncommon words may be shared by documents simply 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. CiteSeer is also different from previous citation indexing agents in that the indexing process is completely automatic. CiteSeer autonomously locates, parses, and indexes articles found on the World Wide Web. The publication delay for printed journals and conferences means that CiteSeer has access to articles that are more recent.

4.2.2 Document Distance Measures

CiteSeer uses several methods for document similarity measurement. One very common semantic feature used to gauge document topic similarity is that of word vectors. We have implemented a TFIDF [15] scheme to measure a value of each word stem in each document where a vector of all of the word stem values represent a document. We truncate to the top 20 components for each document for computational reasons, but there is evidence that this truncation should not have a large affect on the distance measures [14]. A string edit distance measure can also be used to determine document similarity. Currently, CiteSeer uses the LikeIt string distance [19] to measure the edit distance between the headers of documents in a database. LikeIt tries to match substrings in a larger string, and common authors, institutions, or words in the title will tend to reduce the LikeIt distance between headers.

Despite their common use, single words (and even phrases) may not always 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 paper's 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 word oriented TFIDF [14] 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 as follows:

1. Use the Identical Citation Grouping (ICG) algorithm on the entire database of documents to get a count (c_i) of how frequently each cited paper i occurs in the database. Take the inverse of these frequencies as a weight for that citation ($w_i = \frac{1}{c_i}$) and store these values as a table in the SQL database. This step only needs to be executed one time once the database has been constructed, and is reused for later queries.

2. Determine the list of citations and their associated weights for document A and query the SQL database to find the set of n documents $\{B_j\} : j = 1 \ldots n$ which share at least one citation with A.

3. For each $j = 1 \ldots n$, determine the relatedness of the document R_j as the sum of the weights of the citations shared with A.

$$R_j = \sum_{(i \in A_i) \cap (i \in B_j)} w_i \tag{2}$$

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

CiteSeer Home Help Suggestions

Query: `document: +recurrent +series`

NEURAL DYNAMICS OF VARIABLE-RATE
SPEECH CATEGORIZATION
by
Stephen Grossbergz, Ian Boardmany, and Michael Cohenz
Department of Cognitive and Neural Systems
and
Center for Adaptive Systems
Boston University

(Details) (Find Similar Articles)

RESONANCE AND THE PERCEPTION OF
MUSICAL METER
Edward W. Large
John F. Kolen
The Ohio State University

(Details) (Find Similar Articles)

[...section deleted...]

276 documents found.

Figure 3: Results of a keyword search on documents in the neural network database.

more highly than a citation made by a large number of documents. Presently, although we have not performed formal performance measures on CCIDF, practically we have found it to be useful, and to perform better than the word vector or LikeIt based automatic similar document retrievers.

Combination of Methods: Although citation based similar document retrieval has proven to be subjectively superior to word vector or LikeIt based retrieval, CiteSeer also combines different methods of document similarity to result in a final similarity distance measure that is hopefully more accurate than any single method alone. We use a weighted sum of document similarity measures as a combined similarity measure, which is computed according to the following combining 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 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 always 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:

$$S_j = w_{WV}WV(A, B) + w_{LI}LI(A, B) + w_{CI}CI(A, B)$$

5. Retrieve the documents with the highest S_j values.

Although this combination scheme is relatively simple, if the weights are properly chosen, logically it will always perform as well as or better than any single similarity measurement method. The limiting case of a weight of 1 for the best performing method shows that this is true. In the future, we intend to explore the use of learning techniques in order to automatically determine the best weights as a function of the particular database in which the combining will be used.

CiteSeer implements this combined, similar document document recommendation mechanism as part of the browsing process. Given a specific target document, The user chooses `(Find Similar Articles)` as seen in Figure 3. The details of the five best documents are returned for display in the Web browser.

5 Conclusion and Future Work

CiteSeer is an assistant agent that automates and enhances the task of finding interesting and relevant research publications on the World Wide Web. Informally, CiteSeer seems to work well as a practical tool which can save researchers a great deal of time and effort in the process of a literature search. However, there are directions in which we intend to further develop this system. Semantic distance measures may be able to assist the recommendation of new interesting documents. As new research papers become available of the web, they can be automatically downloaded and parsed. If a new paper is similar enough to a user-chosen paper of interest, then CiteSeer could notify the user of potentially interesting new research by e-mail. Another direction for future work is the collection of database statistics. For example, the number of times a paper, author, or journal is cited may give some indication of its influence in the academic community. CiteSeer can currently rank papers according the number of citations made to them, however rankings based on authors, journals, etc. are not currently performed. CiteSeer could recommend that the user watch out for interesting new papers from influential authors and journals. As these statistics change over time, this may be an indicator of research trends. Finally, we intend to measure and enhance CiteSeer's performance by using existing bibliographic databases such as the many large BibTeX databases on the Web. BibTeX information is potentially much more accurate than that parsed from a Postscript file, and could be

CiteSeer Home Help Suggestions

Query: document: +recurrent +series

RESONANCE AND THE PERCEPTION OF
MUSICAL METER
Edward W. Large
John F. Kolen
The Ohio State University

This document can be downloaded from: ftp://archive.cis.ohio-state.edu/pub/neuroprose/large.resonance.ps.Z

Abstract: Many connectionist approaches to musical expectancy and music composition let the question of ``What next?'' overshadow the equally important question of ``When next?''. One cannot escape the latter question, one of temporal structure, when considering the perception of musical meter. We view the perception of metrical structure as a dynamic process where the temporal organization of external musical events synchronizes, or entrains, a listener's internal processing mechanisms. This article introduces a novel connectionist unit, based upon a mathematical model of entrainment, capable of phase- and frequency-locking to periodic components of incoming rhythmic patterns. Networks of these units can self-organize temporally structured responses to rhythmic patterns. The resulting network behavior embodies the perception of metrical structure. The article concludes with a discussion of the implications of our approach for theories of metrical structure and musical expectancy. Connection Science, 6 (1), 177 - 208. RESONANCE AND THE PERCEPTION OF MUSICAL METER 1 (Find Similar Items)

Citations made by this document:

Apel, W. (1972) *Harvard dictionary of music (2nd ed.). Cambridge, MA: Belknap Press of Harvard University Press.* (Details)

Allen, P. E. & Dannenberg, R. B. (1989) *Tracking musical beats in real time.* In Proceedings of the 1990 International Computer Music Conference. Computer Music Association. (Details)

Beek, P. J., Peper, C. E. & van Wieringen, P. C. W. (1992) Frequency locking, frequency modula- tion, *and bifurcations in dynamic movement systems.* In G.E. Stelmach and J. Requin (Eds.) Tutorials in motor behavior II. Elsevier Science Publishers B. V. (Details)

Bharucha, J. J. & Todd, P. M. (1989) *Modeling the perception of tonal structure with neural nets.* Computer Music Journal, 13, 44-53. (Details)

Bodenhausen, U. &Waibel, A. (1991) *The Tempo 2 algorithm: Adjusting time delays by supervised learning.* In R. P. Lippmann, J. Moody, & D. S. Touretsky (Eds.) Advances in Neural Information Processing Systems 3. San Mateo, CA: Morgan Kaufman. Bolton, T. (Details)

Carpenter, G. A. & Grossberg, S. (1983) *A neural theory of circadian rhythms: The gated pacemaker.* Biological Cybernetics, 48, 35-59. (Details)

[...section deleted...]

80 citations

Figure 4: Detailed document information in the neural network database.

used to "fill in" information if a simple title match is made. Also, BibTeX files can be used to create a Postscript testing data set to measure CiteSeer's citation parsing performance.

References

[1] Institute for Scientific Information, 1997.

[2] Keycite, 1997.

[3] BALABANOVIC, M. An adaptive Web page recommendation service. In *Proceedings of the First International Conference on Autonomous Agents* (February 1997).

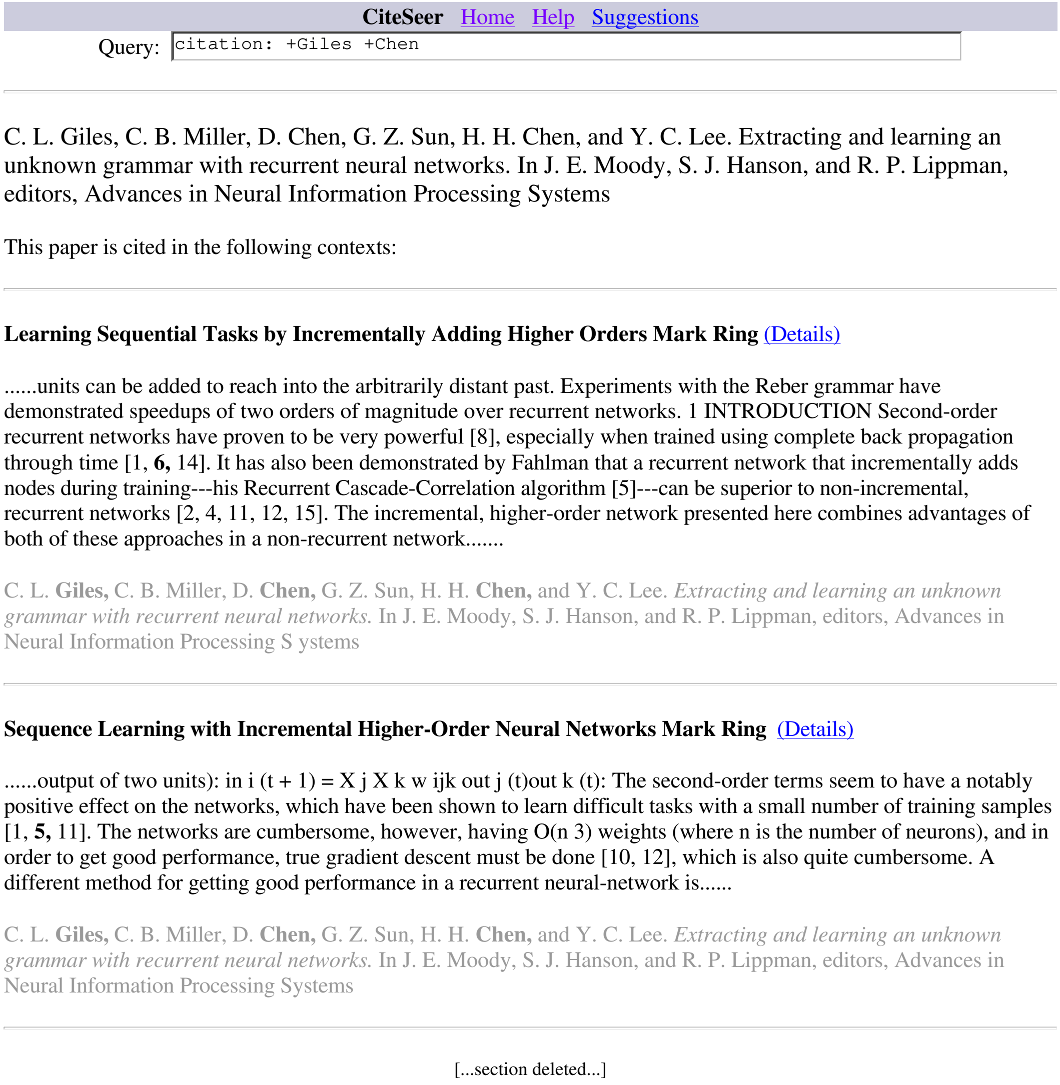
[4] CAMERON, R. D. A universal citation database as a catalyst for reform in scholarly communication. *First Monday* (February 1997).

[5] EDWARDS, P., GREEN, C. L., LOCKIER, P. C., AND LUKINS, T. Exploiting learning technologies for World Wide Web agents. In *IEEE Colloquium on Intelligent World Wide Web Agents,Digest No: 97/118* (March 1997).

[6] GARFIELD, E. The concept of citation indexing: A unique and innovative tool for navigating the research literature. *Current Contents January 3* (1994).

[7] LEVENSHTEIN, V. I. Binary codes capable of correcting spurious insertions and deletions of ones (original in Russian). *Russian Problemy Peredachi Informatsii 1* (January 1965), 12–25.

[8] LOKE, S. W., DAVISON, A., AND STERLING, L. CIFI: An intelligent agent for citation finding on the World-Wide Web. Technical Report 96/4 Dept. of Computer Science, University of Melbourne, 1996.


CiteSeer Home Help Suggestions

Query: citation: +Giles +Chen

C. L. Giles, C. B. Miller, D. Chen, G. Z. Sun, H. H. Chen, and Y. C. Lee. Extracting and learning an unknown grammar with recurrent neural networks. In J. E. Moody, S. J. Hanson, and R. P. Lippman, editors, Advances in Neural Information Processing Systems

This paper is cited in the following contexts:

Learning Sequential Tasks by Incrementally Adding Higher Orders Mark Ring (Details)

......units can be added to reach into the arbitrarily distant past. Experiments with the Reber grammar have demonstrated speedups of two orders of magnitude over recurrent networks. 1 INTRODUCTION Second-order recurrent networks have proven to be very powerful [8], especially when trained using complete back propagation through time [1, **6,** 14]. It has also been demonstrated by Fahlman that a recurrent network that incrementally adds nodes during training---his Recurrent Cascade-Correlation algorithm [5]---can be superior to non-incremental, recurrent networks [2, 4, 11, 12, 15]. The incremental, higher-order network presented here combines advantages of both of these approaches in a non-recurrent network.......

C. L. **Giles,** C. B. Miller, D. **Chen,** G. Z. Sun, H. H. **Chen,** and Y. C. Lee. *Extracting and learning an unknown grammar with recurrent neural networks.* In J. E. Moody, S. J. Hanson, and R. P. Lippman, editors, Advances in Neural Information Processing S ystems

Sequence Learning with Incremental Higher-Order Neural Networks Mark Ring (Details)

......output of two units): in i (t + 1) = X j X k w ijk out j (t)out k (t): The second-order terms seem to have a notably positive effect on the networks, which have been shown to learn difficult tasks with a small number of training samples [1, **5,** 11]. The networks are cumbersome, however, having O(n 3) weights (where n is the number of neurons), and in order to get good performance, true gradient descent must be done [10, 12], which is also quite cumbersome. A different method for getting good performance in a recurrent neural-network is......

C. L. **Giles,** C. B. Miller, D. **Chen,** G. Z. Sun, H. H. **Chen,** and Y. C. Lee. *Extracting and learning an unknown grammar with recurrent neural networks.* In J. E. Moody, S. J. Hanson, and R. P. Lippman, editors, Advances in Neural Information Processing Systems

[...section deleted...]

Figure 5: Detailed citation information in the neural network database.

[9] MENCZER, F. ARACHNID: Adaptive retrieval agents choosing heuristic neighborhoods for information discovery. In *Machine Learning: Proceedings of the fourteenth International Conference* (July 1997), pp. 227–235.

[10] MOUKAS, A. Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem. In *Proceedings of the Conference on Practical Applications of Agents and Multiagent Technology* (April 1996).

[11] PAZZANI, M., MURAMATSU, J., AND BILLSUS, D. "Syskill & Webert: Identifying interesting Web sites". In *Proceedings of the National Conference on Artificial Intelligence (AAAI96)* (1996).

[12] PORTER, M. F. "an algorithm for suffix stripping". *Program 14* (3 1980), 130–137.

[13] SALTON, G. Automatic indexing using bibliographic citations. *Journal of Documentation 27* (1971), 98–110.

[14] SALTON, G., AND BUCKLEY, C. "Term weighting approaches in automatic text retrieval". Tech Report 87-881 Dept. of Computer Science, Cornell University, 1997.

[15] SALTON, G., AND YANG, C. On the specification of term values in automatic indexing. *Journal of Documentation 29* (April 1973), 351–372.

[16] SPERTUS, E. ParaSite: Mining structural information on the Web. In *Proceeding of The Sixth International World Wide Web Conference* (April 1997).

[17] STARR, B., ACKERMAN, M. S., AND PAZZANI, M. Do-I-Care: Tell me what's changed on the Web. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access Technical Papers* (March 1996).

[18] YIANILOS, P. The LikeIt intelligent string comparison facility". NEC Institute Tech Report 97-093, 1997.

[19] YIANILOS, P. N. Data structures and algorithms for nearest neighbor search in general ametric spaces. In *Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms* (1993), pp. 311–321.