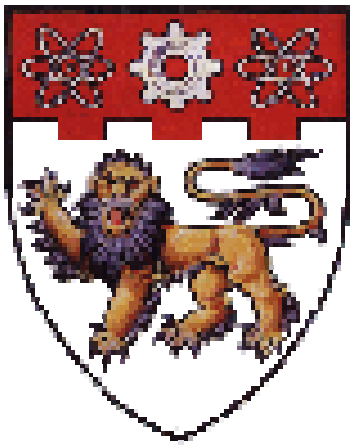
MINING A WEB CITATION DATABASE FOR THE RETRIEVAL OF SCIENTIFIC PUBLICATIONS OVER THE WWW



HE YULAN

SCHOOL OF COMPUTER ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

2001

Mining a Web Citation Database for the Retrieval of Scientific Publications over the WWW

He Yulan

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2001

Acknowledgments

The author's special thanks go to Associate Professor Hui Siu Cheung, her supervisor, for his constant encouragement, thoughtful criticism and suggestions. Without his positive support and invaluable patience, the finish of the thesis is not possible. He has also given the author tremendous help and kept on pushing her to advance to higher stages.

The author also wants to thank Ms. Ding Ying for her help and many useful related materials provided about the citation database research area in the initial stage of this research.

Furthermore, the author wants to thank Mr. Rahul Kaul, an Honors Year student, for his idea and suggestions on the project development.

Finally, the author's gratitude goes to Assistant Professor Alvis Fong, for his review and invaluable comments on the thesis.

And many thanks to everyone else who has helped in one way or another.

Table of Contents

List of Figures

List of Tables

Summary

With the tremendous growth of the information available on the World Wide Web (WWW), significant amounts of time and effort are needed in order to find the relevant information. A lot of commercial search engines are developed to help users to locate the information of their interest by matching their queries against the database of previously indexed documents. However, most search engines only index Web documents, i.e. HTML files, they do not index scientific publications that normally appear in PostScript or PDF (Portable Document Format) formats. Only Google (Google Inc., 2001) has recently announced that it will include PDF sources in its search space. The retrieval of scientific publications over the WWW poses a challenging problem to many researchers in the area of information retrieval.

To tackle this problem, a scientific publication indexing and retrieval system, called PubSearch, has been proposed. It comprises three major components: Citation Indexing Agent, Web Citation Database, and Intelligent Retrieval Agent. The Citation Indexing Agent downloads the scientific literature from the Web, extracts the citation information to form a Web Citation Database. This thesis focuses on investigating the Intelligent Retrieval Agent, which applies data mining techniques on the Web Citation Database to extract useful knowledge for document clustering and author clustering.

For document clustering, the data mining process consists of five steps, namely, feature selection, pre-processing, transformation, document cluster generation, and retrieval. The Kohonen's Self-Organizing Map (KSOM) and Fuzzy Adaptive Resonance Theory (Fuzzy ART) are applied as the mining techniques. Evaluation on

the training and retrieval performance of these two neural networks has been conducted.

For author clustering, the data mining process is based on author co-citation analysis. It consists of six steps, namely, create author co-citation pairs, create raw co-citation matrix, convert into correlation matrix, generate author clusters, display author cluster map, and finally author retrieval. The Agglomerative Hierarchical Clustering (AHC) algorithm is employed to generate the author clusters. The Multidimensional Scaling (MDS) technique is combined with the AHC algorithm to display the author cluster map. Performance evaluation has been carried out on four different similarity measure methods, including the single link, complete link, average link, and Ward's method.

The thesis ends with a brief discussion on the future work, which includes combining the co-citation and co-word analysis, investigating other mining tasks, improving the visualization of the clustering results, and providing online recommendations of new Web publications.

Chapter 1
Introduction

1.1 Scientific Publications over the WWW

In recent years, the World Wide Web (WWW) has become one of the most important media with which people can share information resources. In the past, there was always a considerable time lag between the completion of the research and the availability of the publications. With the WWW, scientific publications can be easily made online. Some research papers are even published on the WWW before they appear in traditional research journals or conference proceedings. In fact, most of the scientific research work published in scholarly journals and conference proceedings are now also provided online in the form of digital libraries (Schatz and Chen, 1996; Levy and Marshall, 1995; Fox *et al.*, 1995; Rao *et al.*, 1995).

For scientific researchers, they can always search around the WWW to keep update on the research trends that are relevant to them. With the help of such information, researchers can concentrate on new research issues and avoid doing research on the topics that have been done and published before. As such, the WWW allows for efficient sharing of information among researchers that is crucial to a successful research environment. However, as most scientific publications are made online by different content providers, they tend to be poorly organized. Thus, it makes the search of relevant research publications difficult and time consuming.

With the enormous amount of publications and information available on the WWW and other networked information sources such as digital libraries, it is necessary to have efficient mechanisms to help researchers to locate related publications accurately and effectively.

1.2 Search Engines

Currently, Web-based search engines such as AltaVista (AltaVista Company, 2000) and Yahoo! (Yahoo! Inc., 2000) have been developed to allow users to specify a query and match it against a database of previously indexed documents. These search engines can be divided into two broad categories: non-robot-driven and robot-driven. For non-robot-driven search engines, the subject directories/trees or hierarchical lists are created manually. Yahoo! is one of the examples of non-robot-driven search engines. In contrast, robot-driven search engines collect information from Web sites on the WWW automatically and create an index for them. There is no human intervention needed. Most commonly used commercial search engines belong to this category. They examine the Web sites in terms of relevance of words contained in each page. In order to do this, they not only index sites submitted by users, but also continuously search the entire Web using specific software programs (so-called *robots*, *spiders* or *crawlers*). Lycos (Lycos Inc., 2000a), Excite (At Home Co., 2000) and HotBot (Lycos Inc., 2000b) are the most popular search engines based on this approach.

However, search engines are insufficient to help searching research publications accurately and efficiently due to the following reasons:

- *Low precision and relevance*. Search engines normally return long lists of ranked documents that users are forced to sift through to find the relevant ones. The results generated are too generalized and not focused on the specific needs of the users.

Moreover, the returned URL (Universal Resource Locator) links may not be available anymore since it may be removed by the provider due to out-dated indices generated by the search engines.

- *Limited search engine coverage*. Search engine coverage relative to the size of the publicly indexed Web has decreased substantially since December 1997. With no engine indexing more than 16% of the publicly indexed Web (Lawrence and Giles, 1999), only a small percentage of the Web pages are indexed by search engines.
- *Unequal access*. Search engines are more likely to index Web sites that have more links to them, more likely to index US Web sites than non-US Web sites, and more likely to index commercial Web sites than educational Web sites.
- *Out of date*. It may take months to index new or modified Web sites by most search engines. It means that users may not get fresh information by using these search engines even if the information is already online. This problem exacerbates with the increasing growth rate of the WWW.
- *Different ranking method*. Search engines produce results according to their own ranking methods and do not tailor to a user's actual needs and interests. For example, Excite uses link popularity as part of its ranking method. Web pages with many links pointing at them are given a slight boost during ranking, since a page with many links to it is probably well regarded on the Internet. Some hybrid search engines, such as those with associated directories, may give a relevancy boost to Web sites they have reviewed. HotBot and Go.com (Disney Enterprises Inc., 2000) will give Web pages an extra boost if search terms appear in their meta tags. In contrast, Lycos does not read them at all. As such, research publications published from various Web sites can get different treatments from different search engines.

- *File format of Web publications*. Currently, most research publications on the WWW are in PostScript or PDF (Portable Document Format) format. They are not accessible through the commercial search engines as most search engines do not index such file formats.

1.3 Intelligent Information Retrieval

Apart from search engines that help users to locate information over the WWW, a number of intelligent information retrieval (IIR) systems (Mladenic and Institute, 1999) have recently emerged to find the truly relevant information to a user's needs. IIR systems are software agents that integrate the techniques of information retrieval and artificial intelligence. Traditional search engines enable users to retrieve potentially relevant Web pages, but unable to provide structural or content-based information. IIR systems require automatic learning and prediction about the relevance of a document's content to the user's information needs. Currently, IIR has been employed by a wide range of applications including Web navigation (Shavlik and Eliassi-Rad, 1998; Pazzani *et al.*, 1996; Rucker and Marcos, 1997), news filtering (Lang, 1995; Konstan *et al.*, 1997), information finding (Hammond *et al.*, 1995; LaMacchia, 1996; Krulwich and Burkey, 1996; Kautz *et al.*, 1997), and information tracking (Yang and Liu, 1999).

Besides the above applications, IIR has also been applied for searching and retrieval of documents over the World Wide Web. WebACE (Han *et al.*, 1998), Bookmark Organizer (Maarek and Ben Shaul, 1996), SONIA (Sahami *et al.*, 1998), and BUS (Shin *et al.*, 1998) have used different clustering techniques to organize Web documents for retrieval. Similar to search engines, these systems are ineffective to

search scientific publications as they focus only on textual Web data format rather than PDF or PostScript format.

Recently, an IIR agent known as CiteSeer (also named as ResearchIndex) (Giles *et al.*, 1998; Bollacker *et al.*, 1998; Lawrence *et al.*, 1999; Bollacker *et al.*, 2000) has been developed for the retrieval of research publications. CiteSeer is an autonomous Web agent for the generation of citation indices. It can automatically generate citation indices from online academic literature in electronic format including PostScript and PDF. In addition, a powerful interactive Web interface is also developed to help finding relevant papers using keyword search or by navigating the links between papers through the citations. CiteSeer differs from conventional citation indexing systems in that it can create more up-to-date citations that are not limited to a pre-selected set of journals and the operation is completely autonomous.

1.4 Citation Database and Retrieval

In published journal articles, there are always some papers or books that are cited as references for the concepts or ideas presented in them. Such citations are used to refer the reader to the relevant papers for further reading on the concepts and ideas that are introduced in the source paper. These cited papers provide a valuable source of information and directives for researchers in the exchange of ideas, the current trends and the future development in their respective fields. A citation index contains the references that a literature cites, linking the source literature to the cited document. Citation indices can be used to identify the research fields or newly emerging areas, analyze research trends, find out the scholarly impact, and avoid duplication of previous works.

Citation database is a data warehouse used for storing citation indices. Some of the information stored in the citation database includes “author name”, “paper title”, and “journal name”. It contains all the cited references (or bibliographies) published with the articles. These cited references reveal how the source paper is linked to the prior relevant research on the assumption that citing and cited references have a strong link through semantics. Therefore, citation index can be used to facilitate the searching and management of information. Some commercial citation index databases such as those provided by the Institute for Scientific Information (ISI) (ISI, 2000) are available over the Web. ISI produces Social Science Citation Index, Arts & Humanities Citation Index, etc. Through these indices, users are allowed to perform searching on cited references from the citation databases.

As discussed earlier, CiteSeer can automatically locate, parse and index scientific publications found on the WWW. It differs from ISI in which ISI generates citation indices for existing periodicals, while CiteSeer is able to capture the most recent “snapshot” of publications on the WWW. Currently, CiteSeer uses Web search engines such as AltaVista, HotBot, and Excite, and heuristics to locate good starting points for searching the WWW. It then downloads PostScript or PDF files of the publications, converts them into text using PreScript from the New Zealand Digital Library project (PreScript, 1998). The converted texts are parsed to extract citations and the context in which the citations are made in the body of the paper. Finally, the extracted information is stored in the citation database.

Both ISI and CiteSeer support citation-based retrieval. ISI provides two types of search: General Search to search for publications by subject term, author name, journal title, or author affiliation, and Cited Reference Search to search for publications that cite the author or publication that the user has specified. CiteSeer also

supports two types of keyword search on citations and indexed publications. The results returned can be ordered by the number of citations or by the publication date. In addition, it also supports Related Document Link that lists the related documents of a selected document. However, both ISI and CiteSeer do not support document clustering retrieval and author retrieval. In this project, we focus on providing these two types of citation-based clustering retrieval techniques using document keywords and author information.

1.5 Data Mining

Data mining, also known as Knowledge Discovery in Databases (KDD) (Fayyad *et al.*, 1996; Fayyad, 1998), has been defined as "the non-trivial extraction of implicit, previously unknown, and potentially useful information from data". It uses machine learning, statistical and visualization techniques to discover the knowledge from large databases. Data mining has been applied to many applications in areas such as analyzing medical outcomes, detecting credit card fraud, predicting customer purchasing behavior, predicting the personal interest of Web users, and optimizing manufacturing processes (Mitchell, 1999). Many data mining tools have also been developed to scour databases for hidden patterns, find predictive information, and allow businesses to make proactive, knowledge-driven decisions.

Different data mining tasks (Fayyad *et al.*, 1996) have been defined. This includes characterization, classification, association, prediction and clustering. Machine learning algorithms are central to these data mining tasks. Some of the most commonly used algorithms in data mining are artificial neural networks, generic algorithms, decision trees, nearest neighbor method, rule induction and data visualization. Many companies provide commercial implementations of these

algorithms (Garofalakis *et al.*, 1999). However, these algorithms have significant limitations (Mitchell, 1999). They typically assume the data contains only numeric or symbolic features but no image feature. They also assume the data has been carefully collected into a single database with a specific data mining task in mind.

1.6 Objectives

The primary objective of this research is to build a citation-based indexing and retrieval system for scientific publications over the WWW. These publications often appear in some academic institution's Web sites in PostScript, PDF or HTML format. Currently, many intelligent systems for the retrieval of Web documents (in HTML format) have been developed, but not on Web scientific publications in PostScript or PDF format. From the previous discussion, citation index can be used as a powerful search tool for scientific literature. This gives us the motivation to generate the citation indices of Web scientific publications and store them into a citation database. Through such citation indices, intelligent retrieval of Web scientific publications is possible.

To achieve this, we have developed a citation-based indexing and retrieval system known as PubSearch. It consists of three major components, namely, Citation Indexing Agent, Web Citation Database and Intelligent Retrieval Agent. The Citation Indexing Agent automatically generates citation indices of Web scientific publications and stores them into the Web Citation Database. The Intelligent Retrieval Agent applies data mining techniques on the Web Citation Database to support intelligent retrieval of Web publications.

This project focuses on applying data mining techniques to the Web Citation Database for scientific publication retrieval. The Web Citation Database has been analyzed to investigate the possible knowledge that could be extracted. As most

researchers are interested in scientific publications within certain research areas, identifying different research areas and authors from the same research area becomes one of the most important knowledge to be mined from the Web Citation Database. Therefore, the mining tasks can be defined as document clustering and author clustering. One possible way to achieve this is through the use of clustering techniques.

Figure 1-1 shows the data mining work of this project, which is listed as follows:

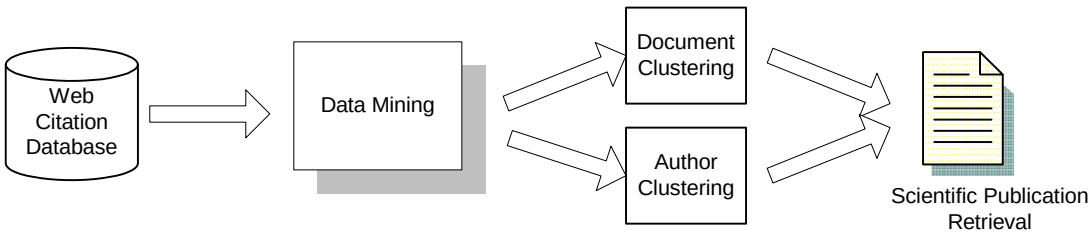


Figure 1-1. Applying data mining on Web Citation Database.

- *Mining for Document Clustering*. Data mining is applied to the Web Citation Database to group Web publications based on keyword similarities between them. Two kinds of neural network techniques, Kohonen's Self-Organizing Map (KSOM) (Kohonen, 1995) and Fuzzy Adaptive Resonance Theory (Fuzzy ART) (Carpenter *et al.*, 1991), are investigated.
- *Mining for Author Clustering*. Author Co-Citation Analysis (McCain, 1990) is incorporated into the data mining process to categorize authors into different research areas from the Web Citation Database. This is based on the assumption that if the frequency of two authors cited by the same publication is very high, these two authors may belong to the same or similar research field.

1.7 Organisation of the Thesis

This chapter gives the background information and motivation of the research work. It briefly discusses the limitations and drawbacks of the commercial Web search engines. Different kinds of intelligent information retrieval agents are also introduced, followed by a brief discussion on citation indexing and data mining. The objectives of the research work are then stated. The rest of the thesis is organized as follows.

Related work on relevant areas of the research is discussed in Chapter 2. It includes a detailed discussion on intelligent agents and various mining techniques that can be applied to the citation database. The proposed approach to mine the Web Citation Database is also given.

Chapter 3 describes the overall structure of the Web publications indexing and retrieval system, PubSearch. The three major components of the system, Citation Indexing Agent, Web Citation Database and Intelligent Retrieval Agent are introduced.

The data mining process for document clustering is discussed in Chapter 4. Two different techniques, Kohonen's Self-Organizing Maps (KSOM) and fuzzy Adaptive Resonance Theory (ART) neural networks, have been implemented in the system. The comparison of these two techniques is also given in this chapter.

Chapter 5 presents the mining process for author clustering. The Author Co-Citation Analysis (ACA) technique is incorporated into the mining process. The performance evaluation on the proposed data mining technique is also presented.

Finally, Chapter 6 concludes the research work and discusses possible areas for future work.

Chapter 2
Related Work

As the project focuses on Web scientific publication retrieval, this chapter first discusses the various intelligent information retrieval agents and their applications. Related work on document retrieval techniques and citation-based retrieval techniques are also reviewed. Then, different data mining tasks and algorithms are presented. As data mining techniques will be applied to the Web Citation Database for document clustering and author clustering, the related clustering algorithms are discussed in detail. Finally, a discussion on the proposed mining techniques to be developed in this research is given.

2.1 Intelligent Information Retrieval Agents

Intelligent Information Retrieval (IIR) agents are software systems deployed on the WWW to help users to search, retrieve, filter and organize information related to their interests. Most IIR systems are used to browse the WWW, retrieve Web documents or find some specific information. Some other IIR systems can also help users to schedule meetings. For example, Calendar Apprentice (Mitchell *et al.*, 1994) connects to a user's electronic calendar and learns the user's scheduling preferences to provide advice to the user for new, unscheduled meeting. Generally, IIR systems or agents can be classified into the following categories: Web navigation, information filtering/categorization, information finding and Web document retrieval.

2.1.1 Web Navigation

This type of Web agents learns the user profiles and discovers Web information that corresponds to users' preferences. WebWatcher (Armstrong *et al.*, 1995) interactively helps users locate desired information by taking keywords from users, suggesting hyperlinks, and receiving evaluation. Wisconsin's Adaptive Web Assistant (WAWA) (Shavlik and Eliassi-Rad, 1998) is another agent that is capable of accepting instructions on the type of information that users are seeking. WAWA then compiles these instructions into a neural network and uses it to guide users to navigate the Web autonomously in the discovery, retrieval and filtering of online information. Another agent system, Syskill & Webert (Pazzani *et al.*, 1996), collects ratings of the explored Web pages from the users and learns their preferences using Bayesian classifier. A Web-page recommendation system, known as Siteseer (Rucker and Marcos, 1997), measures the similarity between bookmark files of different users and groups them accordingly. This will then be used for recommending Web pages for users to navigate.

2.1.2 Information Filtering/Categorization

The IIR systems falling into this category employ various information retrieval techniques to automatically retrieve, filter and categorize Web documents. WebACE (Han *et al.*, 1998) uses clustering algorithms based on graph partitioning to automatically categorize a set of Web documents. Bookmark Organizer (Maarek and Ben Shaul, 1996) combines hierarchical clustering techniques and user interaction to organize Web documents. NewsWeeder (Lang, 1995) is a system for electronic news filtering. It uses the text classification technique to generate a model of users' interests.

2.1.3 Information Finding

These agents search for relevant information using characteristics of a particular domain and possibly a user profile to organize and interpret the discovered information. FAQ-Finder (Hammond *et al.*, 1995) uses a natural-language question-based interface to access distributed text information sources and matches questions from relevant FAQ files against user questions, it will then find the answers to these questions. Internet Fish (LaMacchia, 1996) is another example that also uses a natural-language interface, but instead of finding answers for user questions, it helps users to extract useful information from the Internet. Besides finding information on the WWW, another group of agents searches for expert advice on a given topic for the user. The ContactFinder (Krulwich and Burkey, 1996) agent categorizes bulletin board messages into different topic areas and assists users by referring them to the experts whom can help them. Referral Web (Kautz *et al.*, 1997) constructs the social networks on the Web and searches for the co-occurrence of names in close proximity in any online documents.

2.1.4 Web Document Retrieval

This type of IIR systems retrieves Web documents (normally in HTML format) in different ways. SONIA (Service for Organizing Networked Information Autonomously) (Sahami *et al.*, 1998) has been implemented as part of the Standford Digital Libraries Testbed. It is accessed through the SenseMaker (Baldonado and Winograd, 1997) interface, which allows users to simultaneously query multiple heterogeneous information sources including popular Web search engines, proprietary information database (e.g. DIALOG (The Dialog Co., 2001)) and so on. It then makes use of machine learning methods to extract relevant features from documents through

a multi-tiered feature selection process that is customized to the user's query. Another system, called BUS (Bottom Up Scheme) (Shin *et al.*, 1998), is mainly used for the indexing and retrieval of SGML (Structured Generalized Markup Language) (Herwijnen, 1994) or XML (eXtensible Markup Language) (Light, 1997) documents. The indexing is performed at the lowest level of the given structure and query evaluation computes the similarity at higher level by accumulating the term frequencies at the lowest level in a bottom-up manner.

2.1.5 Discussion

Although there exist so many intelligent agents to help users browse the Web, filter information and retrieve Web document, only a few intelligent systems are available to help users to search for scientific publications. Web scientific publications normally appear in PDF or PostScript formats. As search engines do not index PDF or PostScript files, these Web publications can not be retrieved through them. For most researchers, they can only rely on some simple search tools provided by the content provider for searching and retrieving Web publications. For example, users can make use of the search tools available in IEEE (IEEE, 2000) or ACM (ACM, 2000) digital libraries to locate the required publications. However, only Web publications provided by the content provider can be retrieved.

2.2 Document Retrieval Techniques

The currently available document retrieval techniques can be classified into three broad categories: feature-based approach, database approach, and knowledge representation approach. Popular keyword-based search engines adopt feature-based approach where documents are characterized using feature-based representations.

Systems using the database approach build a fully materialized data warehouse of information in the Web (Chawathe *et al.*, 1994; Atzeni *et al.*, 1997). Knowledge representation approach tries to extract the knowledge from the Web first, and the queries are answered by dynamically accessing the Web based on the extracted knowledge (Arens *et al.*, 1996; Kirk *et al.*, 1995; Etzioni and Weld, 1994).

PubSearch creates the Web Citation Database through its Citation Indexing Agent. It then mines the knowledge from the Web Citation Database for retrieving scientific literature over the Web. Therefore, PubSearch adopts both database and knowledge representation approaches. Database here refers to the Web Citation Database and knowledge extraction requires the use of data mining techniques. Thus, the following sections will discuss the citation-based retrieval techniques and the related data mining algorithms.

2.3 Citation-Based Retrieval

To our knowledge, there are only two systems that support citation-based document retrieval, one is provided by Institute for Scientific Information (ISI) (ISI, 2000), the other is CiteSeer (Giles *et al.*, 1998, Bollacker *et al.*, 1998; Lawrence *et al.*, 1999; Bollacker *et al.*, 2000). As discussed in Chapter 1, ISI maintains a number of citation databases. It provides two types of search: General Search and Cited Reference Search, which are actually simple keyword search. ISI allows users to find related papers. However, the relevance is judged by the citations that are shared by two papers. That is, if there are one or more common citations between two papers, they are considered as related. This method may not reflect the relevance between any two papers accurately, as the citation frequency has not been considered.

CiteSeer supports two types of keyword search on citations and indexed publications. For the citation search, all citations matching the given query along with the context of source papers where the citations occur are retrieved. The results are ordered according to the number of times each paper is cited. When searching the full-text of indexed publications, CiteSeer returns the header for matching publications along with the context of the publication where the keywords occur. Users can order the publications according to the number of citations or by publication date. CiteSeer can also display the related publications. The relatedness is judged using several algorithms. A Term Frequency x Inverse Document Frequency (TFIDF) (Salton and McGill, 1983) scheme is adopted to locate publications with similar words. Distance comparison of publication headers is used to find similar headers. Common Citation x Inverse Document Frequency (CCIDF) is applied to find publications with similar citations.

Different from the above two systems, PubSearch aims to support document clustering as well as author clustering retrieval based on the mining of the knowledge from the Web Citation Database.

2.4 Data Mining Process

Figure 2-1 shows the general data mining process (Fayyad *et al.*, 1996; Mitchell, 1999), which consists of the following five steps:

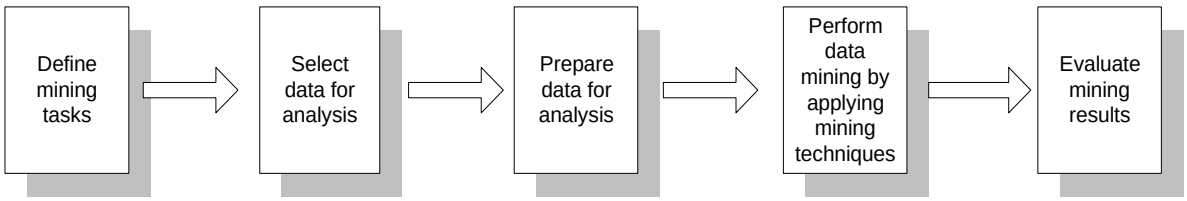


Figure 2-1. Data mining process.

1. *Define mining tasks*. At the beginning of the mining process, the mining tasks or mining goals need to be defined or established.
2. *Select data for analysis*. This includes selecting a dataset or focusing on a subset of variables or data samples on which the knowledge discovery is to be performed.
3. *Prepare data for analysis*. The selected data needs to be pre-processed to remove the noise, replace the missing or unknown values, etc.
4. *Perform data mining by applying mining techniques*. The appropriate mining techniques are investigated and applied to the selected data to extract the hidden relationships, or discover the interesting patterns.
5. *Evaluate mining results*. The mining results need to be interpreted and evaluated to judge their usefulness.

In general, data mining tasks can be classified into two categories (Fayyad *et. al.*, 1996): descriptive data mining and predictive data mining. Descriptive data mining focuses on finding human-interpretable patterns describing the data. Predictive data mining involves the construction of one or a set of models, and attempts to predict the behavior of unknown or future data sets of interest. A data mining system (Han, 1999) may achieve the goals of description or prediction by the following data mining tasks.

- *Generalization and Summarization*. Generalization and summarization provides a description of a behavior from a subset of data. It should cover not only the summary properties such as count, sum, and average, but also the properties on data dispersion such as mean, standard deviation, etc. Attribute-oriented induction (Han *et al*., 1993; Han and Fu, 1996; Michalski, 1983) and data cube (Han, 1997) are the two commonly used approaches to support data generalization and summarization. For example, summarization can be used to compare the sales of

product A and product B and derive an overview of the factors that differentiate the sales of these two products.

- *Association*. Association is the discovery of *association relationships* or *correlation* among a set of items. For example, it can be used to describe which items are commonly purchased with other items in a supermarket. Association approaches often express the resultant affinities in terms of confidence-rated rules such as "80% of all transactions in which beer was purchased also included potato chips". Confidence thresholds can be set to eliminate all but most common trends. The commonly used techniques for mining association rules include Apriori (Agrawal and Srikant, 1994), DHP (Park *et al.*, 1995) and the algorithms for mining the generalized and multi-level association rules (Srikant and Agrawal, 1995; Han and Fu, 1995).
- *Classification*. Classification analyzes a set of training data and constructs a model for each class based on the features in the data. Once an effective classifier is developed, it can be used for better understanding of each class in the database and for classification of future data. For example, in targeted marketing, classification uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. The popular classification methods are decision tree algorithms such as CART (Quinlan, 1986), ID3 (Breiman *et al.*, 1984) and C4.5 (Quinlan, 1993).
- *Clustering*. Clustering is the process of grouping physical or abstract objects into classes of similar objects. Data clustering is to identify clusters embedded in the data, where a cluster is a collection of data objects that are "similar" to one another. Clustering differs from classification in the way that it does not rely on any predefined classes. The clustering process is performed automatically by

clustering algorithms that identify the distinguishing characteristics of the data object and then partition the n-dimensional space defined by the data object attributes along natural cleaving boundaries. Clustering analysis techniques can be based on probability analysis (Fisher, 1987; Fisher, 1995), statistical classification (Cheeseman and Stutz, 1996; Jain and Dubes, 1988), and distance measurement (Agrawal *et al.*, 1993; Faloutsos and Lin, 1995).

- *Prediction*. Prediction predicts the possible values of some missing data or the value distribution of certain attributes in a set of objects. The prediction process involves the finding of the set of attributes relevant to the attributes of interest (e.g. by some statistical analysis) and predicting the value distribution based on the set of data similar to the selected object(s). Usually, tools like regression analysis, generalized linear model, correlation analysis, and decision trees can be used in prediction.
- *Time-series*. Time-series analysis (Han, 1997) is to analyze a large set of time-series data to find certain regularities and interesting characteristics including search for similar sequences or sub-sequences, and mining sequential patterns, trends and deviations.

This research focuses on data mining techniques for clustering. There are two possible ways to apply clustering to the Web Citation Database: document clustering based on keyword similarities, and author clustering based on author co-citation analysis. In the following sections, a more detailed discussion on the techniques that can be used for mining the Web Citation Database for these two types of information will be given.

2.5 Mining Techniques for Document Clustering

Document clustering has traditionally been investigated mainly as a means to improve the performance of search engines by pre-clustering the entire corpus (van Rijsbergen, 1979). Clustering has been extensively studied in the area of literature search (Jardine and van Rijsbergen, 1971; van Rijsbergen, 1974; Croft, 1980), and the common element among clustering methods is a model of word co-occurrence. Documents are categorized into different groups by measuring the degree of overlapping sets of words.

The performance of clustering is directly affected by the clustering methods chosen. There is a large number of clustering algorithms. All of them try to maximize the variation between clusters relative to the variation within clusters. They can be broadly divided into two basic categories: hierarchical and non-hierarchical algorithms (Jain and Dubes, 1988; Kaufman and Rousseeuw, 1990). In recent years, many new algorithms for document clustering have also been proposed and implemented. They are quite different from the traditional clustering algorithms and cannot be simply classified as hierarchical or non-hierarchical algorithms.

In this section, we will briefly discuss three categories of clustering algorithms: hierarchical, non-hierarchical, and other document clustering algorithms.

2.5.1 Hierarchical Clustering Algorithms

As the name implies, hierarchical clustering algorithms (Jain *et al.*, 1999) involve tree-like construction process. There are two strategies available for hierarchical clustering. A divisive strategy proceeds by subdividing the initial cluster into smaller groups of documents. An agglomerate strategy proceeds by building the classification tree bottom-up, joining single documents into the clusters with the whole

collection as the tree root at the end. Most hierarchical clustering algorithms are variants of the single link (Sneath and Sokal, 1973), complete link (King, 1967), average link (Sneath and Sokal, 1973), and Ward's method (Ward, 1963). The single link, average link and Ward's method typically take $O(n^2)$ time, while the complete link method typically takes $O(n^3)$ time (Voorhees, 1986).

Within all kinds of the hierarchical clustering algorithms, Agglomerative Hierarchical Clustering (AHC) algorithm is probably the most commonly used. It computes the proximity matrix containing the distance between each pair of patterns first. Each pattern is treated as a cluster in the beginning. Then, it finds the most similar pair of clusters using the proximity matrix and merges these two clusters into one cluster. The proximity matrix is also updated to reflect this merge operation. This algorithm is typically slow when applied to large document collections. It is also sensitive to the halting criteria as the stopping point greatly affects the clustering results. If the stopping point is set too early, there may be too many clusters generated. On the contrary, if the stopping point is set too late, it may end up with too few clusters or even only one cluster.

Some variants of the AHC algorithm have been published in the literature (Zamir and Etzioni, 1998). One of the examples is the Buckshot and Fractionation algorithms introduced in Scatter/Gather (Cutting *et al.*, 1992). Unlike the AHC algorithm that is quite slow, Buckshot and Fractionation are faster, linear time clustering algorithms.

In the following sections, the SONIA (Service for Organizing Networked Information Autonomously) system is taken as an example to illustrate how the AHC algorithm works. The Buckshot and Fractionation algorithms will also be discussed.

2.5.1.1 Agglomerative Hierarchical Clustering (AHC) Algorithm

SONIA (Service for Organizing Networked Information Autonomously) (Sahami, 1998), which has been implemented as part of the Stanford Digital Libraries Testbed, employs machine learning techniques to create dynamic document categorizations based on the full-text of articles. According to the query results, it automatically retrieves, parses and organizes documents into coherent categories. At the same time, the system can save such document organizations into user profiles that can then be used to help classify future query results by the same user.

The clusters are generated as follows. First, the group-average AHC algorithm is used to form an initial set of clusters, which is then further optimized with an iterative method. Both methods rely on the definition of similarity measure of any two documents. The similarity score used in SONIA is based on the expected probabilistic overlapping on words between a pair of documents. The similarity measure between each document and cluster is computed and each document is assigned to the cluster that is the closest. This process is repeated until convergence or some maximum number of iterations is reached.

This algorithm requires some predefined parameters from users, for example, the number of clusters needs to be generated. As such, the result may not be very accurate, as users normally do not know what is the best value for such parameters.

2.5.1.2 Fractionation and Buckshot

The Scatter/Gather system (Cutting *et al.*, 1992) is an example of browsing approach to retrieval process. It uses fast document clustering to *scatter* the collection into a small number of document groups or clusters, and presents short summaries of them to the user. Based on these summaries, the user chooses one or more of the

groups that are potentially interesting. The selected groups are *gathered* together to form a sub-collection. Then, the system operates on this sub-collection and the same process repeats until the groups become smaller and thus in more detail.

Two clustering algorithms are proposed for this approach, Fractionation and Buckshot. Both algorithms use the group-average AHC algorithm in clustering.

Fractionation is used in the initial stage to find the cluster centers over the whole document collection. It finds *k* centers by initially dividing the whole collection into *N/m* buckets of a fixed size $m > k$, where *N* is the total number of documents in the collection and *m* is a randomly chosen number. The group-average AHC algorithm is applied to each of these buckets separately to generate document groups. The generated groups are then treated as individuals to repeat the entire clustering process. This algorithm will obviously suffer the same disadvantage as in AHC, i.e. the need of specifying the arbitrary halting criteria.

Buckshot is used in the refinement stage to further categorize documents in each sub-collection. The idea of the Buckshot algorithm is to choose a small random sample of the documents of size $\sqrt{kn}$, where *k* denotes the desired number of clusters and *n* denotes the number of documents in this sub-collection, then apply the group-average AHC algorithm, and return the centers of the clusters found. This algorithm is not deterministic since random sampling is employed. That is, using this method repeatedly on the same corpus may produce different partitions.

2.5.2 Non-hierarchical Clustering Algorithms

Non-hierarchical clustering algorithms select the cluster seeds first and assign objects into clusters based on the seeds specified. The seeds may be adjusted accordingly until all clusters are stabilized. Non-hierarchical clustering methods

include the K-means algorithm which typically take O(nkT) time (Rocchio, 1966), where k is the number of desired clusters and T is the number of iterations, and the Single-Pass method that takes O(nk) time where k is the number of clusters created (Hill, 1968). These algorithms are faster than the AHC algorithm. One advantage of the K-means algorithm is that it can produce overlapping clusters. Its disadvantage is that the selection of initial seeds may have great impact on the final result. The Single-Pass algorithm also suffers the same disadvantage and another disadvantage is that the effectiveness of the algorithm is dependent on the order in which the documents are processed. Several variants of the K-means algorithm have also been reported in the literature. One of them is the Leader Clustering algorithm (Hartigan, 1975), which selects the initial partition by assigning the first data item to a cluster and considers the next data item by measuring the distance between the new item and the existing cluster centroids. This process repeats until all data items are clustered.

The interests in artificial neural networks (Hertz *et al.*, 1991) started more than a decade ago. In general, the application of neural networks may be applied in areas that are characterized by (1) noise, (2) poorly understood intrinsic structure, and (3) dynamic nature. These characteristics are common to document clustering process. As there is no satisfying way to represent text documents so far, noise is inevitably imposed. The poorly understood intrinsic structure is due to the fact that it is impossible to know the content of every document in the document collection. Finally, the new documents are kept on adding into the document collection which makes it dynamic.

Some of the well-known examples of artificial neural networks used for clustering include Kohonen's Self-Organizing Map (KSOM) (Kohonen, 1995) and Adaptive Resonance Theory (ART) (Grossberg, 1986) models. Competitive learning

is required in these neural networks, however, the learning or weight update procedures are quite similar to those in some classical clustering approaches. For example, KSOM is essentially a stochastic version of K-means clustering method (Jain *et al.*, 1999). What distinguishes KSOM from K-means is that in addition to the closest cluster, the neighboring clusters are updated as well. The learning algorithm in ART models is similar to the Leader Clustering algorithm (Moor, 1988).

KSOM is one of the most popular unsupervised tools for ordering high-dimensional statistical data in the way that similar input items are mapped close to each other. KSOM can be used to display a colourful map of topic concentrations, which can be further explored by the user by drilling in to browse the specific topic. However, KSOM does not allow gradual learning, that is, whenever there is a new document added into the document collection, the learning process needs to be performed on the whole enlarged document collection again, which makes this algorithm computationally expensive. This trade-off between continued learning and buffering of old memories is called the stability-plasticity dilemma (Grossberg, 1986). ART was introduced in 1986 to solve this problem.

The following sections will introduce the KSOM and ART neural network models.

2.5.2.1 Kohonen's Self-Organizing Maps

The Kohonen's Self-Organizing Maps (KSOM) (Kohonen, 1995) basically converts patterns of arbitrary dimensionality into the responses of one- or two-dimensional arrays of neurons. The feature mapping can be thought of as a non-linear projection of the input pattern space on the neurons' array that represents features. Learning within self-organizing feature maps results in finding the best matching

neuron cells that also activate their spatial neighbors to react to the same input. After learning, each input causes a localized response having a position on the neurons' array that reflects the dominant feature characteristics of the input. The KSOM neural network training algorithm (Kohonen, 1995) is shown in Figure 2-2.

For each encoded input vector X, do step 1 to step 2 and repeat the same process for the whole training set for user-defined number of times:

1. Obtain the similarity measure between the input vector and the weight vectors of the output nodes, and compute the winner output node as the one with the shortest normalised Euclidean distance.

$$|X - W_m| = \min\{|X - W_i|\}$$

where W_i is the weight vector.

2. Update weight vectors as:

$$\Delta W_i(t) = \alpha(N_i, t)[x(t) - w_i(t)] \qquad \text{for i} \in \text{N}_\text{m}(\text{t})$$

where $\text{N}_\text{m}(\text{t})$ denotes the current spatial neightbourhood, α is a positive-valued learning function, $0 < \alpha(N_i, t) < 1$. The function α can be respresented as:

$$\alpha(N_i, t) = \alpha(t)\exp(-\frac{|r_i - r_m|}{\sigma^2(t)}) \qquad \text{for i} \in \text{N}_\text{m}(\text{t})$$

where r_m and r_i are the position vectors of the winning cell and the winning neighbourhood nodes respectively, and $\alpha(\text{t})$ and $\sigma(\text{t})$ are suitable decreasing functions of learning time t.

Figure 2-2. KSOM neural network training algorithm.

Some systems use KSOM neural network for information retrieval. One of the notable examples is the WEBSOM system (Honkela *et al.*, 1997; Honkela *et al.*, 1998; Kohonen *et al.*, 2000), which uses KSOM to group documents according to the words that they contain. The WEBSOM method has been used to organize articles from the Usenet newsgroups as illustrated at the WWW address http://websom.hut.fi/websom/. The interface of the system shows the cluster map and allows the user to zoom in any

particular clusters to see more detailed information by clicking the map image with the mouse.

There are three main phases in WEBSOM: the pre-processing of the input, the formation of the word category map and the formation of the document map. Before applying KSOM to the document collection, some non-textual information and the words that occurred below certain threshold are removed from the newsgroup articles. On the word category map, words are clustered into an ordered set of word categories. Related words fall into the same or nearby categories. Then, the documents can be encoded as word category histograms with the aid of word category map. Related words in the same or similar categories would contribute similarly in the document encoding process. Finally, these encoded documents are presented as inputs to KSOM, which organizes them by unsupervised learning. After the learning process, the document clusters shown as different regions of the document space can be visualized on the document map.

Currently, WEBSOM allows users to move on the document map, zoom in and view the contents of the nodes. WEBSOM may also be used for content-directed document search. The position of the retrieved document on the document map provides a starting point for exploring related documents in the nearby areas. However, WEBSOM does not provide explicit listing of related documents. It still depends on the users to manually explore the relevant documents. On the other hand, the WEBSOM method is only applicable to textual documents. Scientific literature in PDF or PostScript format need to be converted into text files first before the WEBSOM method can be applied to them.

2.5.2.2 Fuzzy Adaptive Resonance Theory

Basically, an Adaptive Resonance Theory (ART) network consists of two layers of units. The units contained in the first layer receive input from the outside world. Therefore, this layer is referred to as the feature representation field. The units contained in the second layer are used to represent the clusters of the input data. This layer is referred to as the category representation field. Weighted connections exist between every unit of these two layers.

The ART family consists of a series of models, including ART, ART1, ART2, ART3, Fuzzy ART, etc. The first ART model was developed by Grossberg (Grossberg, 1986) to solve the problem of trade-off between continued learning and buffering of old memories (i.e. stability-plasticity dilemma). ART1 is the binary version of ART, which can stably learn to categorize binary inputs presented in an arbitrary order (Carpenter and Grossberg, 1987a). ART2 (Carpenter and Grossberg, 1987b), ART3 (Carpenter and Grossberg, 1990) and Fuzzy ART (Carpenter *et al.*, 1991) have been developed to handle multiple valued pattern vectors, that is, either binary or analog data. ART2 and ART3 may be computationally inefficient due to the need to iteratively normalize patterns (Carpenter and Grossberg, 1987b; Carpenter and Grossberg, 1990). The Fuzzy ART model is based on the fuzzy logic computations. It is capable to categorize arbitrary collections of arbitrarily complex analog input patterns effectively.

In our research, the input to the ART network is document vector, which contains analog data. Therefore, only ART2, ART3, and Fuzzy ART could be considered. Properties of learning for Fuzzy ART have been reported in (Carpenter *et al.*, 1991). One of the important properties is the short training time. Hence, Fuzzy ART is chosen to be the ART network model used in the research.

Fuzzy ART incorporates computations from fuzzy set theory into ART1 systems by replacing the non-fuzzy intersection operator ($\cap$) that describes ART1 dynamics (Carpenter and Grossberg, 1987a) by the fuzzy AND operator ($\wedge$) of the fuzzy set theory.

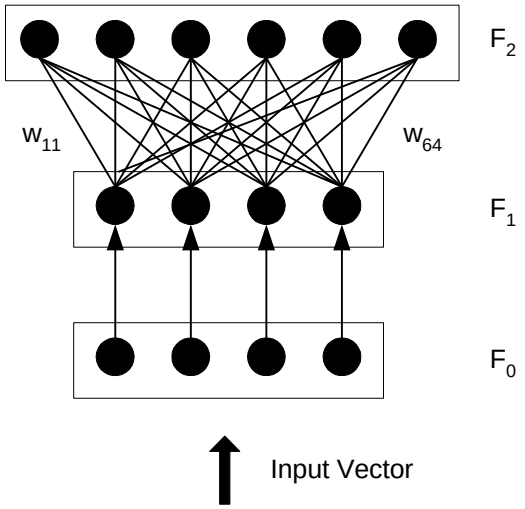


Figure 2-3. Architecture of Fuzzy ART neural network model.

Figure 2-3 illustrates the architecture of the Fuzzy ART neural network model. Each Fuzzy ART system includes a pre-processing field F_0, an input field F_1, and a category representation field F_2. F_0 modifies the current input vector, while F_1 receives both bottom-up input from F_0 and top-down input from F_2. If the original input vector is M-dimensional, then the F_1 field will have 2M nodes (as the original input vector needs to go through the complementary coding in the F_0 field), and the F_2 field will have N nodes, where N represents the maximum number of categories that the F_2 field can accomodate. Each of the N category nodes in the F_2 field have 2M connections with the F_1 field. Each node j in F_2 field has an associated vector W_j, distributed along the connections from that node to all the nodes in the F_1 field. W_j is called the weight vector for the j^{th} cluster. Initially, before the learning occurs, all the

weights in the vector W_j have the value 1 and each category node is said to be uncommited. Only after a category node codes its first input, it becomes committed.

For each encoded input vector, do step 1 to step 4 and repeat the process for the whole training set until no change in the weights of the network.

1. Normalise the input vector to prevent category proliferation. The complemented coded $F_0 \rightarrow F_1$ input **I** is a 2M-dimensional vector:

   $$\mathbf{I} = (a, a^c) = (a_1, \ldots, a_M, a_1^c, \ldots, a_M^c)$$

   where $a_i^c = 1 - a_i$ for $i \in [1, M]$.

   A complemented coded input is automatically normalized, it is because

   $$|I| = |(a, a^c)| = \sum_{i=1}^{M} a_i + (M - \sum_{i=1}^{M} a_i) = M$$

2. For the input **I** and F_2 node j, the choice function T_j is defined by

   $$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|}$$

   where the fuzzy intersection $\wedge$ is defined by $(P \wedge Q)_i \equiv \min(p_i, q_i)$ and where the norm $|\ |$ is defined by $|P| \equiv \sum_{i=1}^{M} |p_i|$.

   The system makes a category choice where at most one F_2 node can become active at a given time. The index J denotes the chosen category, where

   $$T_J = \max \{T_j : j = 1 \ldots N\}.$$

   The output vector **Y** of the field F_2 is set as $y_J = 1$ and $y_j = 0$ for $j \neq J$.

3. Resonance occurs if the match function of the chosen category meets the vigilance threshold, i.e.

   $$\frac{|I \wedge w_J|}{I} \geq \rho$$

   then the weight vector w_J is adjusted according to the equation:

   $$w_J^{(new)} = \beta(I \wedge w_J^{(old)}) + (1 - \beta) w_J^{(old)}$$

   Otherwise, mismatch reset occurs, where the value of the choice function T_J is set to 0. The search process continues until a chosen category meets the vigilance criteria.

Figure 2-4. Fuzzy ART neural network training algorithm.

There is no distinction between training mode and retrieval mode in the ART model. The training can be performed continuously whenever a new input is presented to the Fuzzy ART network. The training algorithm of the Fuzzy ART neural network is shown in Figure 2-4.

Three parameters determine the dynamics of a Fuzzy ART network, a choice parameter $\alpha > 0$, a learning rate parameter $\beta \in [0, 1]$, and a vigilance parameter $\rho \in [0, 1]$. α affects the bottom-up inputs that are produced at the F_2 nodes according to the input pattern presented at F_1. β controls the adjustment of the wight vector W_j. The vigilance threshold level indicates how close an input must be to a stored cluster to provide a desirable match. The higher the vigilance threshold, the more precise the documents are categorised.

2.5.3 Other Document Clustering Algorithms

Recently, many other document clustering algorithms have been proposed, including Suffix Tree Clustering (Zamir and Etzioni, 1998), Supervised Clustering (Aggarwal *et al.*, 1999), and Word Clustering (Slonim and Tishby, 2000).

Suffix Tree Clustering (STC) is a linear time clustering algorithm based on identifying the phrases that are common to groups of documents as opposed to other algorithms that treat a document as a set of unordered words. STC has three logical steps: (1) document "cleaning"; (2) identifying base clusters using a suffix tree; and (3) combining the base clusters into clusters. Document "cleaning" stems each word, strips the non-word tokens and marks the sentence boundaries. The identification of base clusters can be viewed as the creation of an inverted index of phrases for the document collection. This is done using a data structure called a suffix tree (Gusfield,

1997). Documents may share more than one phrase. In the last step, the base clusters with high overlap in their document sets are merged.

In contrast to most other clustering algorithms, which are unsupervised clustering, Supervised Clustering assumes that a pre-existing sample of training documents with the associated classes is available in order to provide the supervision to the categorisation of the whole document collection. A set of seeds which are representative of the defined classes are identified and they serve as the starting points of the subsequent clustering process. The subsequent clustering process is independent of any further supervision. The number of clusters is maintained by either merging two clusters if the similarity of seeds of them is higher than a predefined threshold or by discarding a cluster if the number of documents in the corresponding cluster is less than a predefined value.

The Word Clustering method is quite different from all the other clustering algorithms as it incorporates the information bottleneck method (Tishby *et al.*, 1999). It has two stages. First, word clusters are extracted based on the distribution of the documents in which they occur. In the second stage, the original representation of the documents, the co-occurrence matrix of documents versus words, is replaced by a much more compact representation based on the co-occurrence of the word-clusters in the documents. Using this new document representation, the same clustering procedure for word clusters can be re-applied to obtain the desired document clusters. The main advantage of this algorithm lies in a significant reduction of the inevitable noise of the original co-occurrence matrix of documents versus words, due to its very high dimension. But the time complexity of this algorithm is $O(|X|^3)$, where X is the number of documents to be processed, as double-clustering procedure is used, which is not suitable for very large datasets.

2.5.4 Discussion

Although there are many different document clustering techniques as discussed in this section, the techniques to be used depend very much on the nature of the problem and the final goal to achieve. In this project, the source documents in the Web Citation Database are represented by the keyword vectors with words extracted from their citations and the clustering is based on word co-occurrence between document vectors. We aim to find the clustering algorithm that is fast in terms of computational complexity. As discussed earlier, non-hierarchical algorithms usually have linear time complexity, which makes them the best candidates to satisfy the speed requirement.

Another factor to consider is on how to present the clustering results to the user intuitively. The Kohonen's Self-Organising Map (KSOM) algorithm used in the WEBSOM system can generate a map display easily. Such maps provide a visual overview of the whole document collection with similar documents located close to each other. As mentioned earlier, the KSOM algorithm is closely related to the classical K-means clustering algorithms (Forgy, 1965; Lloyd, 1982, Linde *et al.*, 1980). KSOM without the neighbourhood function is equivalent to K-means. In KSOM, the computational complexity of constructing the mapping function is $O(M^2)$, where M denotes the number of model vectors (Kaski *et al.*, 1998). But in the special case where the ratio between the "width" of the neighbourhood and the size of the map is fixed, the computational complexity is only O(M). So there is a trade-off between the computation time and the size of the map M that determines the resolution of the mapping. In this project, the size of the map will be limited to be 10×10 (i.e. a maximum of 100 clusters will be generated), therefore, the computation time can still be considered as linear.

KSOM has been applied for information retrieval in many different ways. Lin (Lin, 1997) used KSOM to form a map based on titles of scientific documents. Rauber and Merkl (Rauber and Merkl, 1999) had developed a SOMLib digital library based on KSOM. The WEBSOM method is different from the above in the idea of applying KSOM algorithm twice: first for word category analysis and second for document maps creation based on the first analysis. Although the word category analysis can help to group the synonymous or interrelated words together, the "too good" generalising ability of the word category map may result in inaccurate clusters. The reason behind is that words having similar role in sentences while describing different themes may be mapped to the same node.

However, KSOM is not suitable for the dynamic environment like the Web Citation Database as the database is kept on changing by adding in new scientific publications. To cater for the new information, re-clustering needs to be done on the whole database. It is very time consuming and inefficient. Therefore, Fuzzy ART will be the better choice as it allows continuous learning and does not require re-learning for the whole document collection. However, ART nets are order-dependent, that is, different categories are obtained for different orders in which the data is presented to the net. Also, the size and number of clusters generated by ART depend on the value chosen for the vigilance threshold, which is used to decide whether a pattern is to be assigned to one of the existing clusters or a new cluster.

In this project, both KSOM and Fuzzy ART algorithms are investigated as data mining techniques used in document clustering.

2.6 Mining Techniques for Author Clustering

There are two types of co-citation relationships: one is based on documents while the other is based on authors. Document co-citation analysis was first introduced by Small and Griffith (Small and Griffith, 1974) in mid-1970s as one of the major quantitative techniques in science studies to map the structure and dynamics of scientific research. The major function of document co-citation analysis measures the number of documents that have cited a given pair of documents together. This is referred to as co-citation strength. It reflects the frequency of items being cited over time. The patterns revealed by document co-citation generally agree with patterns of direct citation, but differ significantly from bibliographic coupling patterns. Two papers are bibliographically coupled if they cite one or more reference(s) in common, while document co-citation is a relationship between cited documents. White and Griffith later introduced Author Co-Citation Analysis (ACA) in 1981 where the co-citation relationship is based on authors (White and Griffith, 1981). That is, if the frequency of two authors cited by the same paper is high, then there may exist certain relationship between these two authors.

Using document co-citation approach, it is possible to analyze the contribution of a specific document to a given research field. That is, this approach focuses on very fine granularity. Author co-citation approach has a major advantage over the document co-citation approach in that it can identify the intrinsic inter-connectivity links that might be missing using document co-citation approach (Chen and Carr, 1999). For example, one author may write several different articles but all on the same concept. These articles are cited by different source papers. From the citers' point of views, all these articles should be equivalent as they talk about the same concept. But document co-citation analysis would treat them as different articles and fail to recognize the

fundamental connections among them. On the other hand, author co-citation relationships are dynamic and reflect the evolution, decline, and merger of research fields. Hence, only author co-citation relationships will be examined in this research. In this section, the document co-citation method will be briefly reviewed while the author co-citation method will be discussed in detail.

2.6.1 Document Co-Citation

The algorithm developed by Small and Griffith (Small and Griffith, 1974) takes citation index as initial input. For all the documents in the citation index, cited frequency is first calculated by measuring the number of times a document was cited. Documents with cited frequency above a certain threshold are kept for further processing. Co-citation pairs are then generated together with their associated frequency of co-occurrence. Finally, all the documents are clustered by having at lease one document of the co-citation pair in common. For example, a pair (A, B) is selected at the beginning, all the pairs that contain A or B are added to the cluster. This process repeats until no pairs that contain a common document with those in the cluster. Then, another pair is selected from the remaining author co-citation pairs to form a new cluster. The whole process repeats until all the pairs belong to a cluster.

The fundamental premise of document co-citation analysis is that “the greater the number of times that a pair of documents are cited together, the more likely that they are related in content” (Bellardo, 1980). However, it should be understood that two different documents might be cited together in a third document for a wide variety of reasons. They may contribute to the same kind of knowledge or theory, or they may represent totally opposite theories. Even so, document co-citation analysis is still considered as a powerful method to achieve document clustering since it provides a

way to measure the scholarly dependency upon previous works. It also shows the coherence of the literature and changes over time in an intelligible way.

2.6.2 Author Co-Citation

Author Co-citation Analysis (ACA) (White and Griffith, 1981) provides a method for tracing the intellectual structure in science studies. It is based on the frequency with which any works by an author is linked to any works by another author in a third and later works. A common sequence of steps on ACA is given in Figure 2-5:

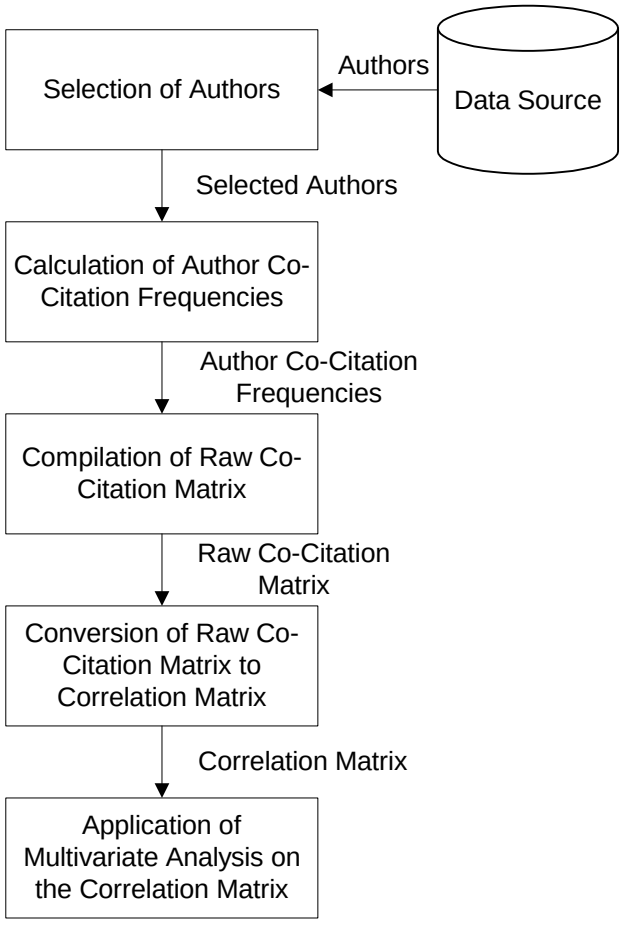


Figure 2-5. Procedure of Author Co-Citation Analysis (ACA).

1. *Selection of authors*. When the data store is huge, only a small sample of authors is selected for analysis. In order to reflect the actual scholarly relationship of the original data store as much as possible, authors should be selected from various

research fields. This author set will define the scholarly landscape that being mapped.

2. *Calculation of author co-citation frequencies*. This step calculates the number of co-occurrence of any two different authors from the selected sample.
3. *Compilation of raw co-citation matrix*. The author co-citation frequencies will form a two dimensional matrix with identically ordered authors' names on the rows and columns. The co-citation matrix is computed using a citing frequency threshold to reduce the set of potential candidates for clustering.
4. *Conversion of raw co-citation matrix to correlation matrix*. The raw data matrix is converted to a matrix of proximity values, which indicate the degree of similarity of author-pairs. The Pearson correlation coefficient (Johnson, 1988) has been used as the measure of similarity in many ACA studies. Another method of transformation is to rank and order the co-citation frequencies by row and assign the appropriate mean rank value to each cell.
5. *Application of multivariate analysis on the correlation matrix*. The values in the correlation matrix define the degree of similarity. The higher the positive correlation, the more similar the two authors are from the perceptions of citers. Multivariate analysis is applied to the correlation matrix to generate the clusters.

Step 5 is the most crucial part of ACA. In general, three approaches can be used for multivariate analysis: Cluster Analysis, Multidimensional Scaling and Factor Analysis (McCain, 1990). Different from the document clustering based on the word co-occurrence, these techniques are based on the computation of similarities among co-citation patterns of each author. The following sub-sections will discuss these three techniques in details.

2.6.2.1 Cluster Analysis

Two most popular approaches to Cluster Analysis are Agglomerative Hierarchical Clustering (AHC) (Everitt, 1986) and Iterative Partitioning algorithms (Jain and Dubes, 1988). They belong to hierarchical clustering methods and involve a tree-like building process as mentioned in Section 2.5.1. The difference between them is on "bottom-up" approach versus "top-down" approach. In the AHC algorithm, clusters are built from the bottom with individuals or groups of individuals gradually joining to form clusters, while in the Iterative Partitioning algorithm, a collection of all individuals are split from the top to the bottom and the process iterates until the desired number of clusters is reached. The AHC algorithm is commonly used in ACA research work.

The AHC method can be applied to the correlation matrix. Authors are joined to the existing cluster or two clusters are merged together based on the similarity values between different author-pairs. SPSS-X (SPSS Inc., 2000) provides a clustering program that implements a variety of AHC procedures, including the single link, complete link, average link and Ward's method. Most ACA researchers use the complete link or Ward's method as they consistently perform well on co-cited author data in terms of providing interpretable results (McCain, 1990).

2.6.2.2 Multidimensional Scaling

Multidimensional Scaling (MDS) (Green *et al.*, 1989) is used to create visual displays or maps from correlation matrices, so that the underlying structure within a set of objects can be studied. Authors who are heavily co-cited appear close to each other in the multidimensional space. Authors with many links to others tend to be in central positions, while authors who are weakly linked will be placed in the periphery.

In this way, central and peripheral research specialization can easily be shown (Kruskal, 1977).

The multidimensional scaling programs used in ACA include MDSCAL, TORSCA (two stand-alone programs), and ALSCAL (Computer Software, SPSS-X) (SPSS Inc., 2000). The input to MDS can be a similarity matrix or correlation matrix. The output of MDS is a display of points, usually mapped into two or three dimensions. Points representing authors with high similarities will be placed closely together in the "intellectual space" while points representing authors with high dissimilarities will be placed farther apart. The main purpose of MDS is to maintain the same relationship between the original data as much as possible in two or three dimensions.

2.6.2.3 Factor Analysis

Essentially, Factor Analysis (Gorsuch, 1983) attempts to "explain" the inter-relationships observed among the original variables through the creation of a much smaller number of "derived" variables or factors. In ACA, a factor is interpreted by the subset of authors loading on it, i.e. making substantial contributions to its construction. Every author loads on (contributes to) every factor, and the interpretation or definition of each new factor is based on those authors with high loadings. The strength of inter-correlation among the factors may also reveal subject-related linkage between authors. The stopping rule decides the number of factors extracted. In SPSS-X, the stopping rule is implemented as the sum of the squared loadings on the factor is less than 1. The advantage of factor analysis is its ability to demonstrate the breadth of contributions by authors who load substantially on more than one factor.

2.6.2.4 Discussion

For Factor Analysis, as every author contributes to every factor and the definition of each new factor is based on those authors with high loadings. The computation complexity will be $O(N^f)$, where N is the number of authors being analyzed and f is the number of factors finally extracted. As discussed in the previous section, the Agglomerative Hierarchical Clustering (AHC) method has the computation complexity $O(N^2)$, where N is the number of clusters generated. It is much faster than the Factor Analysis and the algorithm is simpler to implement. Thus, the AHC method will be used for author clustering in this project.

Multidimensional Scaling (MDS) is mainly used to create the visual display of points. The points in the map represent the individual author placed according to the inter-author similarities. The map requires some manual processing to group points to form clusters. On the other hand, deciding on the number of dimensions is an important issue for MDS. Normally, the input correlation matrix is mapped into two or three dimensions for easy analysis. But this may result in a very poor, highly distorted representation of the original correlation matrix. In this research, we tend to present the author cluster map with each point representing an author in a two-dimensional map. Therefore, MDS is applied in our project to obtain the XY-coordinate of each author from the correlation matrix.

To our knowledge, there are no systems which can automatically generate author clusters based on the author co-citation analysis. Most researchers in information studies field still rely on some statistical tools, like SPSS, to do the analysis on author co-citation data. The author co-citation raw matrix needs to be generated either manually or using other software first before feeding into the statistical tools. The results produced are not well-clustered author groups. Researchers

still need to manually group authors according to the positions of the authors in the output display. So the whole process depends heavily on human interpretation. In PubSearch, author clusters are generated automatically and no human interpretation is required at all.

2.7 Summary

From the survey of various intelligent agents available nowadays, it can be observed that there is a lack of searching tools specifically designed for scientific publications retrieval. Citation indexing has been proved to be an appropriate way to index scientific literature. In this research, Web Citation Database is generated to store the citation index information. Data mining is applied to the Web Citation Database to extract the useful knowledge for scientific publication retrieval. Two data mining tasks are defined based on the structure and attributes of the Web Citation Database, which are document clustering and author clustering. Various techniques for document clustering as well as author clustering are reviewed and compared. After evaluating the different types of clustering algorithms, KSOM and Fuzzy ART are selected as the appropriate methods for document clustering, while AHC and MDS are found to be the suitable methods for author clustering.

Chapter 3
Web Citation Indexing and Retrieval System

As discussed in Chapter 1, this research aims to develop a Web Citation Indexing and Retrieval System known as PubSearch which consists of three major components: Citation Indexing Agent, Web Citation Database and Intelligent Retrieval Agent. In this chapter, the system overview of PubSearch is first given. As the thesis focuses on the Intelligent Retrieval Agent, the Citation Indexing Agent will only be briefly reviewed. The Web Citation Database will be mined for document and author clustering for supporting information retrieval in the Intelligent Retrieval Agent. Thus, it is necessary to have an in-depth understanding of the structure of the database, which will be described here.

3.1 System Overview of PubSearch

Figure 3-1 shows the system overview of the PubSearch system. Citation Indexing Agent generates the Web Citation Database. It first downloads scientific publications from the Web. There are two ways to do this. The first method is similar to CiteSeer in that Citation Indexing Agent uses Web search engines (like AltaVista, Excite, HotBot, etc.) to search for pages that contain keywords such as “paper”, “postscript”, “publications”, etc. Another way is to download the papers from the Web sites that are specified by the users. The downloaded papers are then parsed to extract the citations. The citation indices are generated subsequently and stored in the Web

Citation Database. Intelligent Retrieval Agent applies data mining techniques to the Web Citation Database to discover the hidden relationships among the research publications and explore the useful knowledge that will help to improve the efficiency and effectiveness of the retrieval.

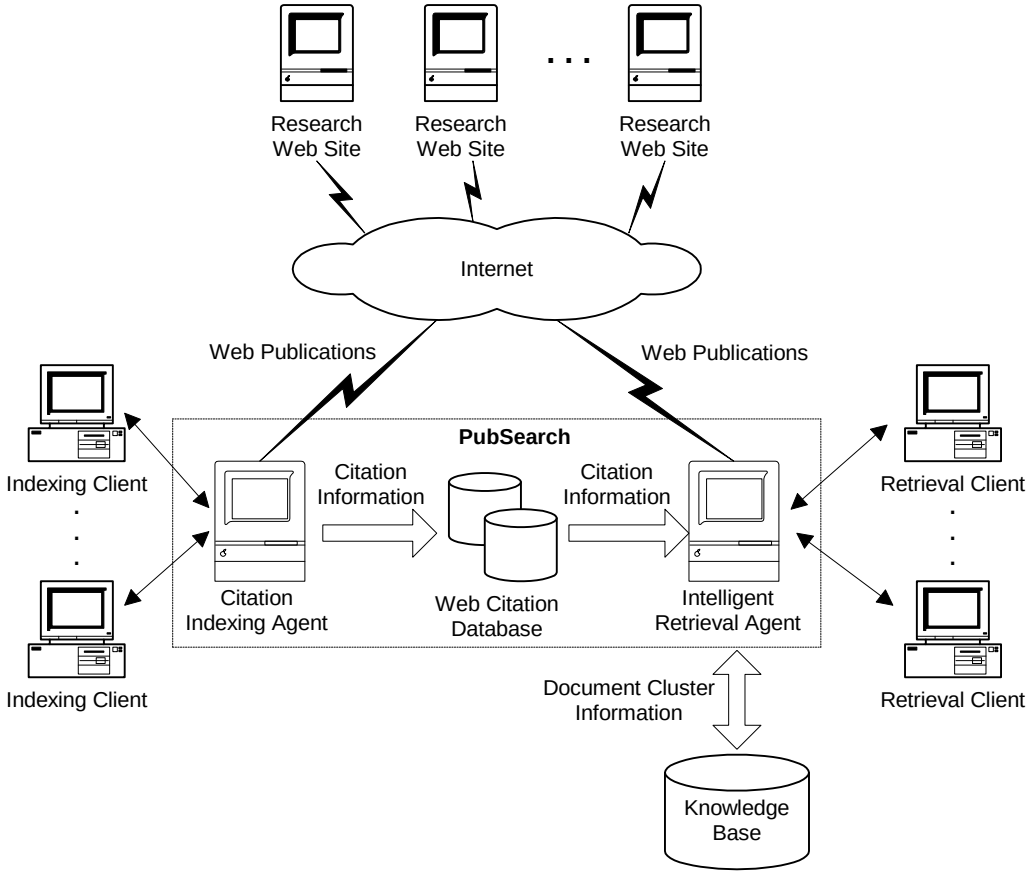


Figure 3-1. System overview of PubSearch.

In the PubSearch system, users interact with two types of clients, one is the Indexing Client and the other is the Retrieval Client. If the user is only interested in the papers from certain Web sites, he/she can use the Indexing Client to specify these Web sites and the frequency that the Citation Indexing Agent needs to monitor them periodically. Otherwise, the Citation Indexing Agent only uses the default monitoring frequency, i.e. seven days to track the Web sites found by the Web search engines. If there are any new papers, the Web Citation Database will be updated accordingly. To retrieve scientific publications, user's query is presented to the Retrieval Client, which

will then pass it to the Intelligent Retrieval Agent to get the search results and display to the user.

3.2 Citation Indexing Agent

Citation indexing is one possible way to index Web scientific publications. It allows the navigation backward in time through the list of cited articles and forward in time to predict what are the newly emerging research fields by identifying the research trends. This makes citation index to be a powerful tool for scientific literature search. Generally, citation indexing techniques can be classified into two broad categories, namely, manual indexing and automatic indexing.

3.2.1 Manual Indexing

Currently, most citation indices for scientific publications are created manually. Some existing commercial citation databases (including *Science Citation Index*) provided by the Institute for Scientific Information (ISI) (ISI, 2000) and the legal database offered by the West Group (WestGroup, 2000) depend on human preparation of the information. There are only two different search strategies provided by ISI, one is through keyword indices, and the other is through citation indices. As the citation indexing process requires human processing, the ISI citation databases are biased because the selection of the items to be indexed depends on the management decisions of ISI (Cronin and Snyder, 1997).

3.2.2 Automatic Indexing

Opposed to manual citation indexing, automatic citation indexing requires no human processing. All the processes, namely, documents acquisition, parsing, and

citation extraction, are done automatically. Currently, the only automatic citation indexing system can be found on the Web is CiteSeer (Giles *et al.*, 1998; Bollacker *et al.*, 1998; Lawrence *et al.*, 1999; Bollacker *et al.*, 2000).

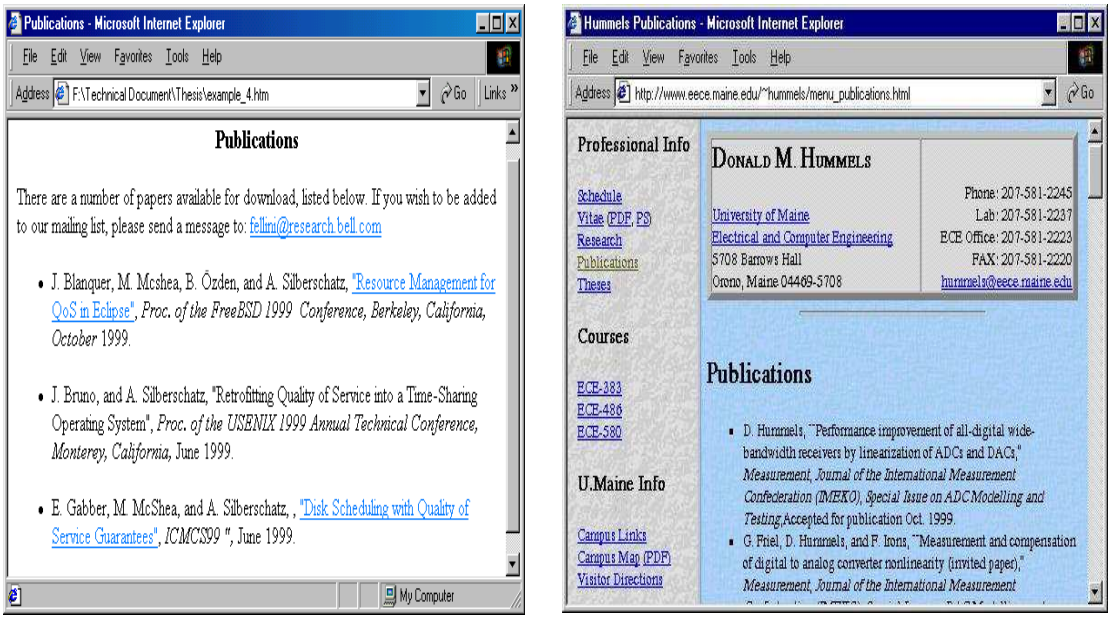
CiteSeer uses Web search engines (such as Alta Vista, HotBot, and Excite) and heuristics to locate papers. For example, CiteSeer can search for pages which contain the keywords “publications”, “papers”, “postscript”, etc. After getting the URL address as the starting point, CiteSeer can then visit that URL location to download PostScript or PDF files, and convert them into text using PreScript from the New Zealand Digital Library project (PreScript, 1998). The converted text files are first verified to be valid research documents by checking the existence of the reference or bibliography section. Then, the following information are extracted: URL address of the downloaded file, title and author block, abstract, introduction, citations, and full-text. Once the reference section is identified, the individual citations are extracted and parsed into the following fields: title, author, year of publication, page numbers, and citation tag. The citation tags, for examples, [3], [Giles 97], and “Fayyad 96”, refer to the information used in the body of the document to help readers to locate the citation. The citation tags can be used to extract the context of citations from the document body.

3.2.3 PubSearch Approach

It can be observed from the above that the automatic citation indexing method is much more efficient. It does not require human processing, hence, no bias will be caused. In PubSearch, the automatic citation indexing method will be used. The Citation Indexing Agent is currently under development by another student (Ho, 2000).

PubSearch is different from CiteSeer, which also applies the automatic citation indexing method, in the following ways:

- *Indexing*. Different from CiteSeer, in addition to using Web search engines to locate papers, PubSearch is also designed to cater for personal interests. It allows users to specify their interested Web sites. Figure 3-2 gives two examples of Web sites which list the publications from a research institution and an individual researcher. Based on these Web sites, PubSearch can download all the papers from them, and extract the citation information to form the citation database.



(a) (b)

Figure 3-2. Examples of two publication Web sites.

- *Monitoring*. CiteSeer uses a combination of Web search engines, Web crawling, and mailing list monitoring to continuously search for new scientific publications 24 hours a day. We doubt the efficiency of such method as continuous updating needs to consume expensive resources and computational power. While for PubSearch, it provides periodic updating and monitoring functionality, which can update the citation database automatically if there are any new papers added to the monitored Web sites. The updating frequency can be specified by the user, which

could be three days, one week or one month. An example of the monitoring interface is given in Figure 3-3.

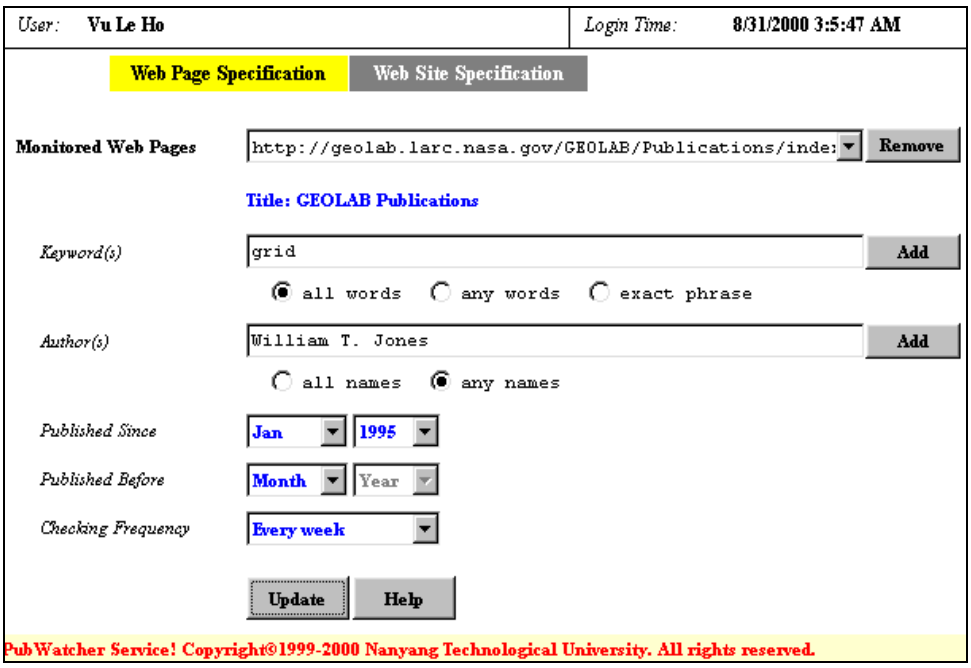


Figure 3-3. Monitoring interface.

3.3 Web Citation Database

Figure 3-4 shows the relationships of the two major tables created in the Web Citation Database. They are the SOURCE and CITATION tables. The attributes of each table are also given. The SOURCE table stores the information of source papers while the CITATION table stores all the citations extracted from the source papers. Most attributes of these two tables have the same data definitions such as the paper title, author name, journal name, journal volume, journal issue, page number, and the year of publication. URL_link is the Web URL address of the corresponding document. With this field, full-text access is possible. The "Paper_ID" of the SOURCE table and the "Citation_ID" of the CITATION table are the primary keys in these two tables respectively. The "No_of_citation" of the SOURCE table is the

number of references contained in the source paper. The "Source_ID" of the CITATION table links to the "Paper_ID" of the SOURCE table to identify the source paper that cites the particular publication stored in the CITATION table.

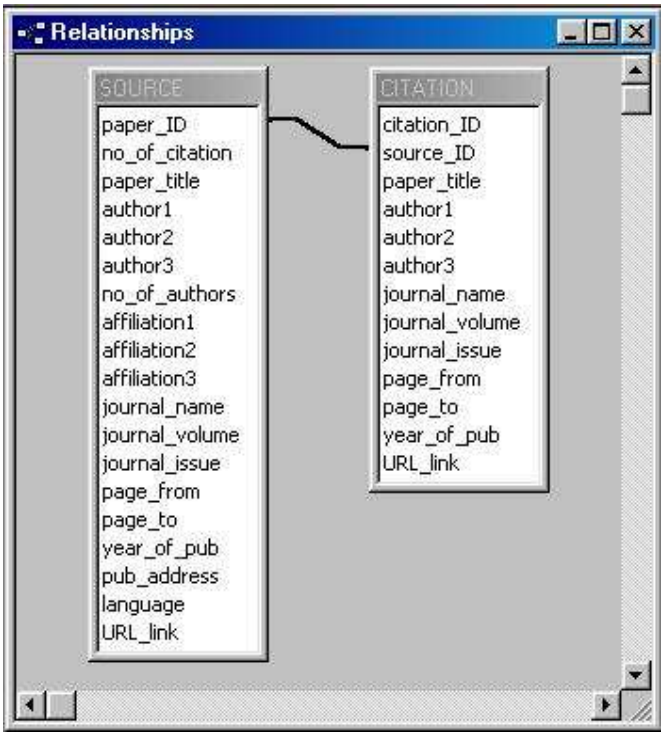


Figure 3-4. Database structure of the Web Citation Database.

Table 3-1 lists the description of all the fields in the Web Citation Database. The *Description* column briefly describes each field while the *Table* column indicates which table contains that particular field. Most fields in the CITATION table are similar to those in the SOURCE table. It should also be noted that for all the papers, only the first three authors are stored in the citation database. This is based on the assumption that the fourth author onwards contributes little to the paper.

Table 3-1. Data field description of the Web Citation Database.

Field Name	***Description***	***Table***
paper_ID	Unique identifier of the paper	SOURCE, CITATION
no_of_citation	Number of citations in the paper	SOURCE
paper_title	Source paper title	SOURCE, CITATION
author1	The first author name	SOURCE, CITATION
author2	The second author name	SOURCE, CITATION
author3	The third author name	SOURCE, CITATION
no_of_authors	Number of authors	SOURCE
affiliation1	Affiliation of author1	SOURCE
affiliation2	Affiliation of author2	SOURCE
affiliation3	Affiliation of author3	SOURCE
journal_name	Journal name	SOURCE, CITATION
journal_volume	Journal volume	SOURCE, CITATION
journal_issue	Journal issue	SOURCE, CITATION
page_from	Starting page number of the paper	SOURCE, CITATION
page_to	Ending page number of the paper	SOURCE, CITATION
year_of_pub	Year of publication	SOURCE, CITATION
pub_address	Publisher address	SOURCE
language	Language of the paper	SOURCE
URL_link	URL address of the paper	SOURCE
citation_ID	Unique identifier of the cited paper	CITATION
source_ID	The identifier of the source paper that cites the paper which is referred by “citation_ID”	CITATION

An example of records stored in the SOURCE and CITATION tables is illustrated in Figure 3-5. Records in the SOURCE and the CITATION tables have many-to-many relationships. That is, one source paper from the SOURCE table may cite multiple papers in the CITATION table. While one record in the CITATION table may be cited by more than one source paper in the SOURCE table. The example shows that both source papers 1068 and 1124 cite the same paper entitled “A simple blueprint for automatic Boolean query processing” written by Salton. On the other hand, the source paper 1068 cites papers by Salton and by Harter at the same time.

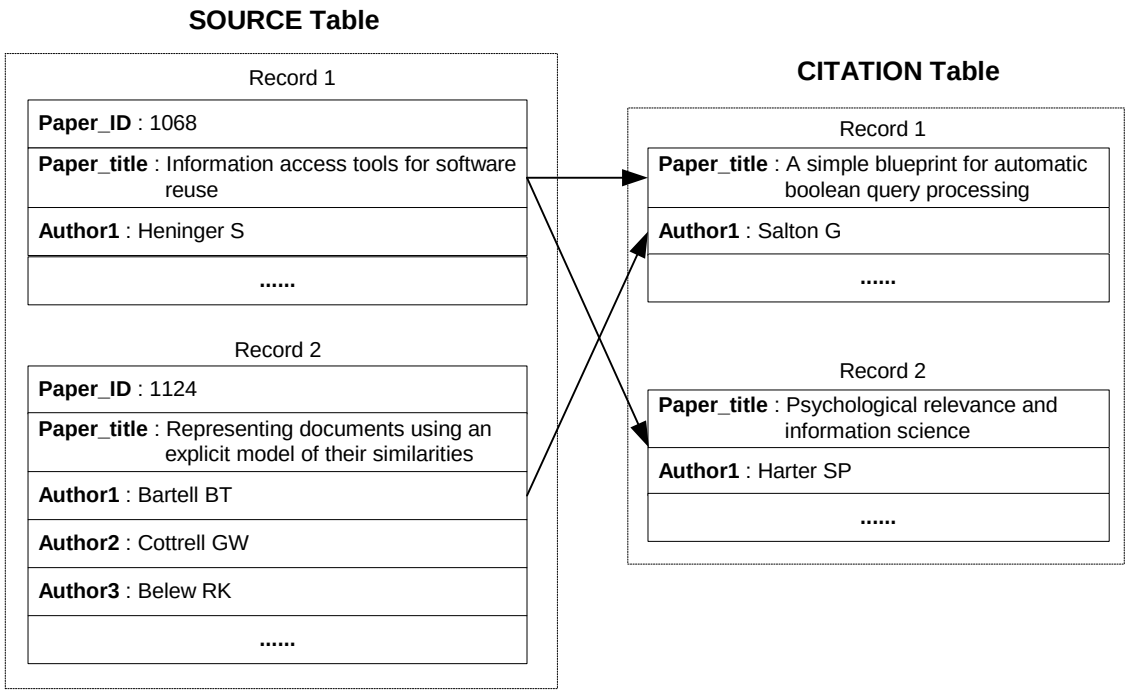


Figure 3-5. Example of records stored in the SOURCE and CITATION tables.

The Web Citation Database contains rich information that can be mined for scientific publication retrieval. The traditional document clustering technique measures the similarity between documents by counting the overlapping keywords within them. Similarly, in the Web Citation Database, the keywords contained in the cited paper titles can also be used to measure the similarity of source papers. On the other hand, author information can be analyzed to reveal the relationships between the citing and cited authors. Therefore, by examining the Web Citation Database, the paper title and author name fields will be used for the mining purposes.

3.4 Intelligent Retrieval Agent

The Web Citation Database stores bibliographic data on published journal articles. It can be used as a source of information for discovering the hidden relationships between papers and authors. Most of these relationships cannot be readily seen and require some statistical methods and analysis. Some of the functions that can

be supported using the citation database for information processing are listed as follows:

1. *Search for publications based on author, title, abstract, and keyword*. This kind of search is widely implemented in many information retrieval systems. Simple Boolean search algorithm is applied to refine the search.
2. *Search for papers belonging to a particular field or category*. A large category may be sub-divided into smaller categories. Search may be refined to a smaller category to provide a more specific search of interest to users.
3. *Create a list of experts within a research field*. Researchers can be categorized into different groups according to their research interests.
4. *Track the research activities of a certain researcher*. The publications by a particular researcher can be displayed in chronological order, which allows users to track his/her research activities.
5. *Determine the impact of a publication on the research field by the frequency it has been cited by other publications*. However, this is only relevant for publications that have existed for some time. New publications will need time to be cited by future publications.
6. *Discover the future trends of specific research fields*. The historical data on the change of research fields can be gathered to predict the trends of research areas.

PubSearch will investigate the first three functions. Search for publications based on author, title, or keyword can be realized by incorporating simple Boolean search mechanism into the system. For the second and third functions, document clustering and author clustering will be performed on the Web Citation Database to extract cluster information, such that users can search papers belonging to a particular field or get a list of experts within the same research area. The fourth function can be

achieved by performing simple author name search on PubSearch. The returned results will be a list of publications ordered by "year_of_publication", which can give users a rough idea on that author's research activities. For the fifth function, it can be realized by deriving the impact factor of the publication. For the last function, to predict the future trends, text mining techniques (Dörre *et al.*, 1999) need to be performed. However, the accuracy of the research trend is difficult to evaluate. Hence, it is not the focus of our research.

In this research, document clustering is the primary mining task to be performed. To achieve this, the input records need to be pre-processed to form the document vectors. Due to the storage limitation, no full-text of source papers are stored in the Web Citation Database. Therefore, the traditional approach of forming the document vector by extracting the keywords from the body text of the document cannot be adopted here. However, citation information can be used to judge the relevance of documents as cited articles are picked by the authors as related documents (Giles *et al.*, 1998). The more the two source documents share the same citations, the higher the possibility they belong to the same research area. Thus, keywords of the cited paper titles are used as the feature factors to represent the source document.

The citation information gives us the idea on how the two authors' research interests are related through other papers that cite their works. As such, author clustering is another mining task that can be performed. As discussed above, the "source_ID" of the CITATION table links to the "paper_ID" of the SOURCE table, that is, all the records with the same "source_ID" in the CITATION table are publications cited by the same source paper. Based on this idea, author co-citation pairs can be created, which can then be used for author clustering.

3.5 Test Citation Database

During the development of the Intelligent Retrieval Agent, the Web Citation Database is not available as it is under the construction of another student who focuses on the Citation Indexing Agent (Ho, 2000). Therefore, a test citation database, which follows the same structure of the Web Citation Database, is used. This database is created by downloading the publications from 1987 to 1997 in Information Retrieval (IR) field of the Social Science Citation Index from the Institute for Scientific Information's Web site, which includes all the journals on Library and Information Science. A total of 1,466 IR related papers were selected from 367 journals with 44,836 citations. The two tables, SOURCE and CITATION, were set up based on these IR papers.

3.6 Summary

In this chapter, the overview of the Web citation indexing and retrieval system known as PubSearch is given. The three major components of PubSearch, namely, Citation Indexing Agent, Web Citation Database and Intelligent Retrieval Agent, are introduced here. Citation Indexing Agent is only briefly reviewed, as it is not the focus of this research. The structure of the Web Citation Database is discussed. It contains two major tables, SOURCE and CITATION. Two mining tasks are investigated, they are document clustering based on citation paper titles and author clustering based on author names. These mining tasks are performed by the Intelligent Retrieval Agent.

To our knowledge, CiteSeer is the only indexing system available for searching Web scientific publications. However, PubSerach differs from CiteSeer in three ways: indexing, monitoring and retrieval. During the indexing process, CiteSeer only uses

Web search engines to download papers, it may end up with many repeated or irrelevant Web sites as one of the main drawbacks for search engines is the long list of irrelevant search results. Besides using Web search engines, PubSearch also allows users to specify their interested Web sites to download papers. In this way, PubSearch tailors to users' interests. CiteSeer monitors the new scientific publications continuously 24 hour a day. It consumes expensive resources and computational power. PubSearch provides periodic monitoring functionality. The retrieval method in CiteSeer is quite restricted. The first query to CiteSeer must be keyword search. CiteSeer then returns a list of citations or a list of indexed articles matching the query. The literature can be browsed by following the citation links. PubSearch provides more flexible ways for retrieval as it incorporates intelligent techniques. With this, it allows users to conduct keyword search as well as author search. The results will be the publications or authors that belong to the same research area even if the keywords or author names do not appear in the user specified query.

Chapter 4
Data Mining for Document Clustering

This chapter discusses a data mining process that mines the Web Citation Database for document clustering. The proposed data mining techniques are based on two different neural network models, namely, Kohonen's Self-Organizing Maps (KSOM) and Fuzzy Adaptive Resonance Theory (Fuzzy ART). The implementation as well as the evaluation of the training and retrieval performance of the two models will be discussed.

4.1 Data Mining Process

Knowledge discovery can be either directed or undirected (Berry and Linoff, 1997). Directed knowledge discovery is characterised by the presence of single target field whose values are to be predicted in terms of the other fields of the database. In undirected knowledge discovery, there is no target field. It is normally used to identify patterns or recognise relationships in the data. Mining cluster information from the Web Citation Database is considered as undirected knowledge discovery process as it reveals the hidden relationships from the citation database.

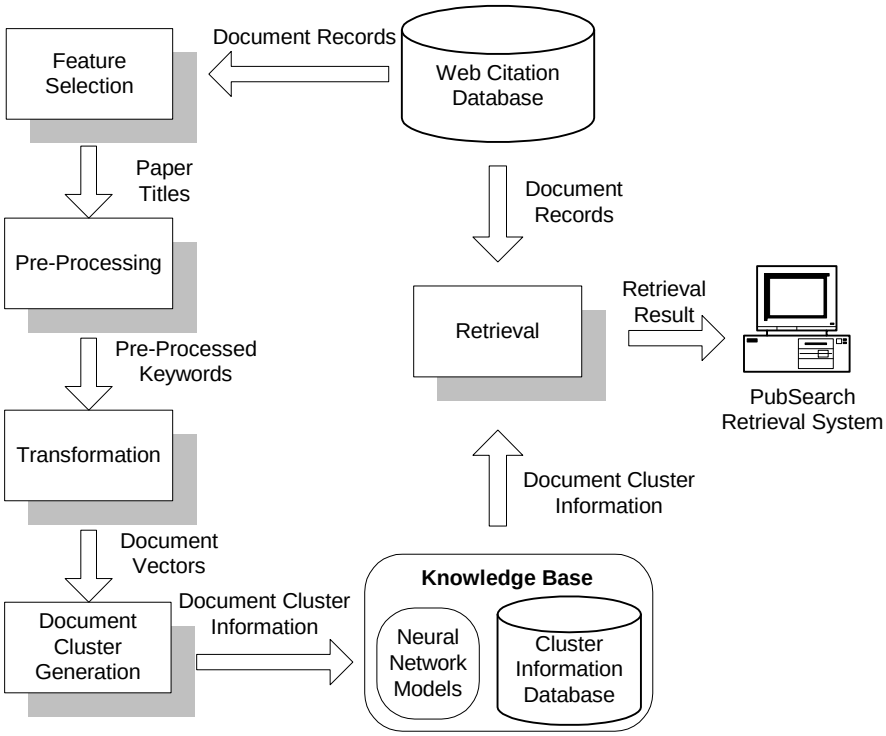


Figure 4-1. Data mining process for document clustering.

Figure 4-1 shows the data mining process for document clustering. It consists of the following five steps:

1. *Feature Selection*. The paper titles of the citation records in the Web Citation Database are extracted as feature factors to represent the document vectors.
2. *Pre-Processing*. The extracted paper titles are pre-processed for subsequent processing. It includes tokenization, stemming, and stop word removal.
3. *Transformation*. The extracted keywords are converted into document vectors. Random projection method (Kaski, 1998) is used to reduce the dimensionality of the document vectors.
4. *Document Cluster Generation*. KSOM and Fuzzy ART are chosen to be the two data mining techniques to be applied to the Web Citation Database for document clustering.
5. *Retrieval*. After the *Document Cluster Generation* step, the knowledge base is formed, which contains the document cluster information as well as the generated

neural network models. The extracted knowledge is used for document cluster retrieval.

4.1.1 Feature Selection

Salton (Salton and McGill, 1983) discusses clustering approaches using TFIDF vector representations for text data, which is the most popular clustering approach adopted today. Each component of a document vector is calculated as a product of Term Frequency (TF) and Inverse Document Frequency (IDF) as follows:

$$d_i = TF(w_i, d) \times IDF(w_i) = TF(w_i, d) \times \log \frac{D}{DF(w_i)}$$

where d_i is the i[th] element of the vector representation of a document *d*;

$TF(w_i, d)$ is the number of times word w_i occurred in a document *d*;

D is the total number of documents in the document collection;

$DF(w_i)$ is the document frequency, which is the number of documents in which word w_i occurred at least once.

A popular similarity measure, the cosine measure can be used to compute the angle between any two document vectors. Using this method, documents will be classified into different groups according to the distances between them.

TFIDF method is used based on the premise that the full-text of a document is always available such that words can be extracted as the feature factors of the document. However, due to storage limitation, it is not possible to store the full-text for all source documents into the citation database. Therefore, the traditional approach can not be adopted here. Other methods need to be investigated to form the document vector.

After an analysis of the Web Citation Database, we propose to use citation information to judge the relevance of documents as cited papers are picked by the authors as related documents. Therefore, instead of extracting keywords from the full-text of the documents as the feature factors, the keywords are extracted from the citations of source documents. If two documents share the same citation, they will also share the same keywords. In particular, the keywords are extracted from the paper titles of all the citations. For each document, the twenty most frequently occurred keywords will be extracted from its citations by the *Pre-Processing* step. Then, the TFIDF method is adopted to represent the document vector.

4.1.2 Pre-Processing

The *Pre-Processing* step involves text processing techniques, which consist of the following tasks:

- *Tokenization*. This refers to breaking the paper titles selected from the *Feature Selection* step into distinct words. As English words are separated by spaces, therefore, it is easy to extract each single word using spaces as the separator.
- *Stemming*. It converts words into their root forms. For example, “retrieved”, “retrieving”, and “retrieves” will become “retriev”. That is, we simply drop the “ed”, “ing” or “es”.
- *Stop word removal*. Words with weak or no meanings are removed, such as “to”, “the”, “a”, etc. A stop word list is used to identify such words.

The WordNet (WordNet, 2000) library is used to implement these techniques. It has built-in functions to perform the above tasks. The keyword pre-processing algorithm is shown in Figure 4-2. For each document, the first twenty most frequently

occurred keywords are extracted from its citations. A total of 5,487 distinct keywords are extracted from all the citations in the Web Citation Database.

Keyword_Pre-Processing_Algorithm:

1. Sort all the records in the CITATION table of the Web Citation Database in ascending order of the source paper ID.
2. For each record read from the CITATION table, do step 3.
3. If the current record is the first record in the database, go to step 3.1. Otherwise, compare it with the previous record, if they have the same source paper ID, i.e. these two citation records belong to the same source paper, go to step 3.1, else, go to step 3.3.
   - 3.1. Tokenize all keywords from the "paper_title" field of the current record, stem the extracted keywords to their root forms, and remove stop words.
   - 3.2. For every keyword, accumulate the number of occurrence.
   - 3.3. If the current record has a different source paper ID from the previous record, it implies the keywords from the citations of the previous source paper are all extracted. Then, sort the keywords for the pervious source paper based on their occurrence, and take the first twenty most frequently occurred keywords as the feature factors of the previous source paper. Go to step 3.1 to process the current record.
4. Go to step 2 to process the next record read from the CITATION table.

Figure 4-2. Keywords pre-processing algorithm.

4.1.3 Transformation

The *Transformation* step converts documents into vectors before feeding into the neural networks for training. Traditionally, documents are represented using the vector space model (Salton and McGill, 1983) or Latent Semantic Indexing (LSI) (Deerwester *et al.*, 1990). In the vector space model, documents are represented as TFIDF vectors. The major problem of this model is the huge vocabulary in the large collection of free-text documents, which results in a vast dimensionality of the

document vectors. LSI tries to reduce the dimensionality of the document vectors by forming a matrix in which each column corresponds to the vector of a document. Then, the factors of the space spanned by the column vectors are computed using a method called Singular-Value Decomposition (SVD) (Will, 1999). The factors that have the least influence on the matrix are omitted which in turn reduces the dimensionality of the matrix. But this method still incurs expensive computation time.

In this research, the random projection method (Kaski, 1998) is used to reduce the dimensionality of the document vectors without losing the power of discrimination between the documents. This method works as follows: the original document vector is multiplied by a random matrix R which consists of random values, and the Euclidean length of each column of R has been normalised to unity.

Kaski (Kaski, 1998) proved that when the dimension is reduced to *d*, the variance between the document vectors with reduced dimensions and the original document vector is at most *2/d*. If *d* is small, the variance will be big which results in a great loss of the original information. Therefore, *d* needs to be carefully set to retain the original information as much as possible. In this research, the variance between 0.005 to 0.01 is considered as acceptable. That is, *d* will be between 200 to 400. Here, we set this number to 300 in considering the trade-off between the variance and the computational complexity. The original document vector is $1\times5{,}487$, the target document vector is 1×300, therefore, the matrix R should be $5{,}487\times300$. The final document vectors obtained only have 300 dimensions, which increases the learning speed dramatically.

According to Kohonen (Kohonen, 1998), a sparse binary projection matrix R with exactly 5 randomly distributed ones in each column is almost as good as the vector space model. An example of such a matrix is shown in Figure 4-3.

Sparse Binary Projection Matrix
5487x300
(each column consists of five ones)

$$\begin{bmatrix} 1 & 0 & 0 & 1 & \dots & 1 \\ 0 & 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 1 \\ 0 & 0 & 1 & 1 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & 0 & \dots & 0 \end{bmatrix}$$

Figure 4-3. An example of the sparse binary projection matrix R.

Consider the matrix product $Y = X \times R$ where X is the original document vector, Y is the resulting document vector, the computational complexity is *O(nd)*, where *n* and *d* are the dimensions before and after the random projection, which are 5,487 and 300 respectively. The pseudocode for such computation is illustrated in Figure 4-4.

```
for i = 1 step 1 until 300 do y(i) = 0;
for i = 1 step 1 until 300
begin
        for j = 1 step 1 until 5487
        begin
                if R(i, j) = 1
                begin
                        y(i) = y(i) + x(j);
                end
        end
end
```

Figure 4-4. Pseudocode for the computation of dimensionality reduction.

An example of the *Transformation* step is illustrated in Figure 4-5. The first twenty highly occurred keywords are extracted from the citations of the source paper. As there are altogher 5,487 distinct keywords extracted from the whole citation database, each document will be represented as a vector with 5,487 dimensions. The index column of Figure 4-5 represents the position of each keyword in the $1 \times 5{,}487$ vector. The presence of i^{th} keyword will set the i^{th} element in the document vector to be 1. Then, each element of the vector is weighted by TFIDF. The resulting vector is

multiplied by the sparse binary projection matrix to reduce its dimension from 5,487 to 300. Finaly, the vector is normalized before feeding into the neural network.

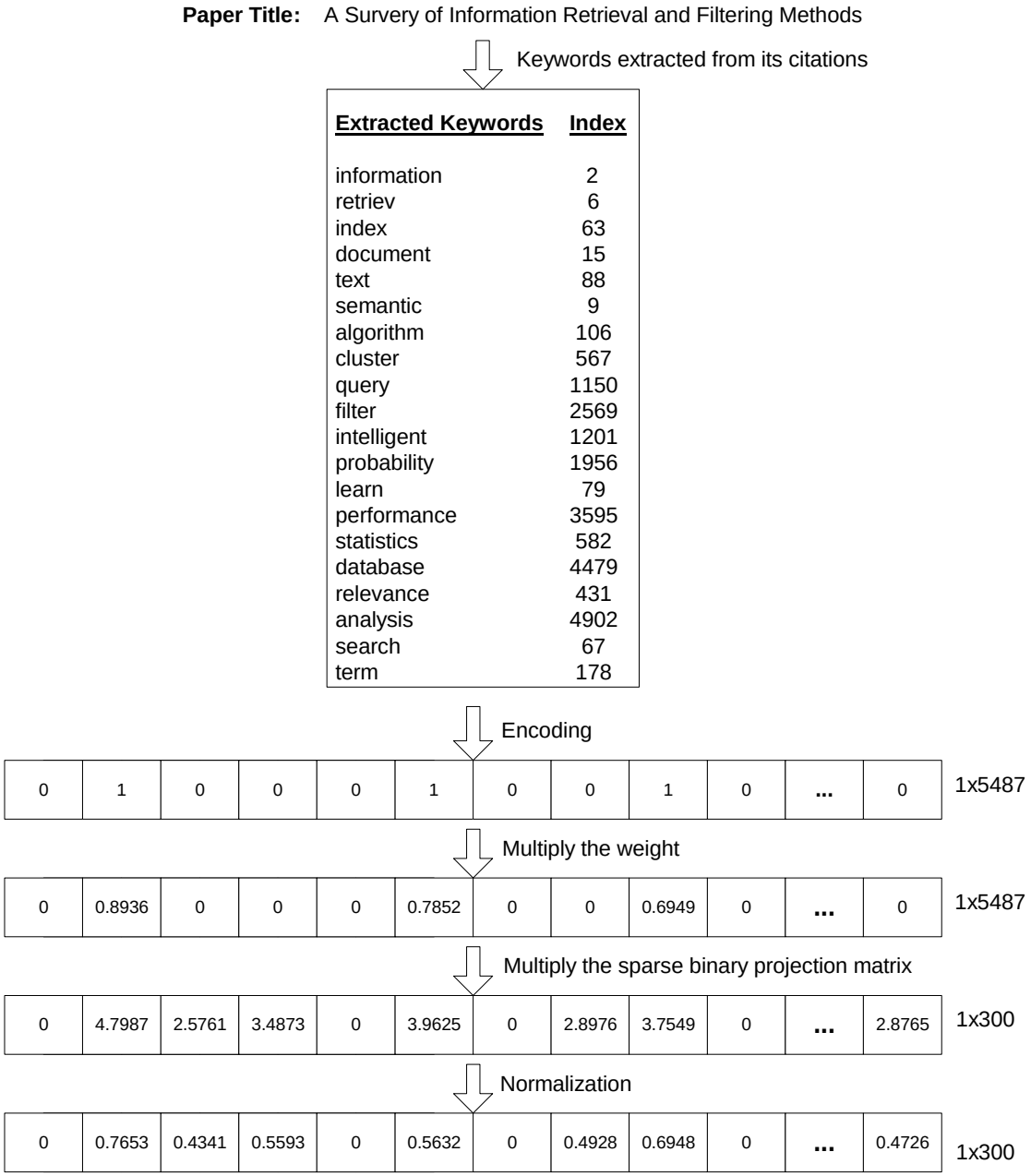


Figure 4-5. An example of the *Transformation* step.

4.1.4 Document Cluster Generation

This is the most crucial step of the overall data mining process. The algorithms used are KSOM and Fuzzy ART neural networks. To categorize the source documents into different clusters, training needs to be conducted for both neural networks. The

training procedures for the KSOM and Fuzzy ART models are not the same due to the differences in the architectures of these two models.

4.1.4.1 Training the KSOM network

Before the encoded document vectors are fed into the KSOM neural network for training, the weights of the network are initialised with the random real numbers within the interval [0, 1]. A total of 1,000 records in the SOURCE table is used as the training set. The performance of the KSOM neural network retrieval depends on the number of clusters generated and the average number of documents within a cluster. However, the decision on the best size of the cluster map remains a non-trivial problem that requires some insight into the structure of the training data. In our implementation, the number of clusters to be generated is set to 100 in order to obtain fast retrieval speed. The initial neighbourhood size is set to be half the number of the clusters. The number of iterations and the initial learning rate are set to 5,000 and 0.5 respectively.

The training algorithm for the KSOM neural network was given in Section 2.5.2.1. During the training, whenever a winner cluster is found, the weights to the winner cluster together with its neighbourhood need to be updated according to the input pattern. The updated weights are stored in a file as the neural network model. During the retrieval stage, no weight updates are performed. This prevents any incoming query from corrupting the “index” stored in the neural networks.

4.1.4.2 Training the Fuzzy ART Network

Three parameters determine the dynamics of a Fuzzy ART network, namely, a choice parameter $\alpha > 0$, a learning rate parameter $\beta \in [0, 1]$, and a vigilance

parameter $\rho \in$ [0, 1]. The choice parameter α affects the bottom-up inputs that are produced at the F_2 nodes according to the input patterns presented at F_1. As short training time is only possible for small values of choice parameter (Carpenter *et al.*, 1991), therefore, α is set to 0.2 in our work. β controls the adjustment of the wight vector W_j. We set β = 1 if j is an uncommitted node and β = 0.5 if j is a committed node. The vigilance threshold level indicates how close an input must be to a stored cluster to provide a desirable match. The higher the vigilance threshold, the more precise the documents are clustered. However, in order to compare the performance of KSOM and Fuzzy ART more closely, the number of clusters to be generated using Fuzzy ART is set as close as possible to the number obtained in KSOM, which is 100. Therefore, the vigilance threshold is set to 0.7 in Fuzzy ART, which has generated 129 clusters. The training algorithm of the Fuzzy ART neural network was given in Section 2.5.2.2.

4.1.4.3 Cluster Information Database

During the training process, the relationship between the input source paper vector and the winning cluster number is saved into the Cluster Information Database. The database contains two major tables, KSOM_OUTPUT and ART_OUTPUT, which store the cluster results from the KSOM and Fuzzy ART neural networks respectively. Table 4-1 gives the data structure of the table KSOM_OUT. It basically stores the information on which cluster the source paper belongs to. The fields “Row_No” and “Column_No” record the position of the cluster in the 10×10 KSOM map. These two fields are used during the display of the document cluster map in the retrieval process.

Table 4-1. Data structure of the table KSOM_OUT.

Field Name	***Description***
Paper_ID	The unique identifier of the source paper.
Cluster_No	The identifier of the cluster that the current source paper belongs to.
Row_No	The row location of the current cluster in the KSOM map.
Column_No	The column location of the current cluster in the KSOM map.

The data structure of the table "ART_OUTPUT" is similar to the table "KSOM_OUTPUT" except that it does not store any "Row_No" or "Column_No" information, as it is not possible to generate the cluster map information.

4.1.5 Retrieval

In the *Retrieval* step, a user's query is submitted to the system. The incoming query is pre-processed, parsed and encoded in a similar way as the *Pre-Process*ing and *Transformation* steps of the data mining process. The encoded query vector is fed into the network to determine which clusters to be activated. This process can be seen as analogous to the use of indices in the document retrieval process. In this scenario, instead of having an index file, the index is encoded in the form of weight distribution in the neural networks. The activated clusters are examined to retrieve the documents within the cluster, which are then ranked based on the closeness to the user's query.

There are three different stages in the *Retrieval* step, namely, pre-processing of user input, document retrieval, and document ranking, which are given in Figure 4-6.

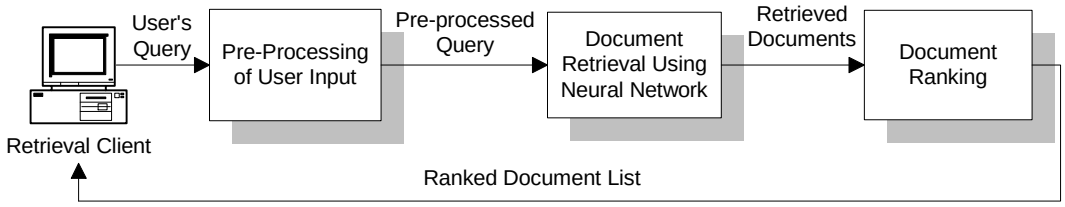


Figure 4-6. The *retrieval* step.

4.1.5.1 Pre-Processing of User Input

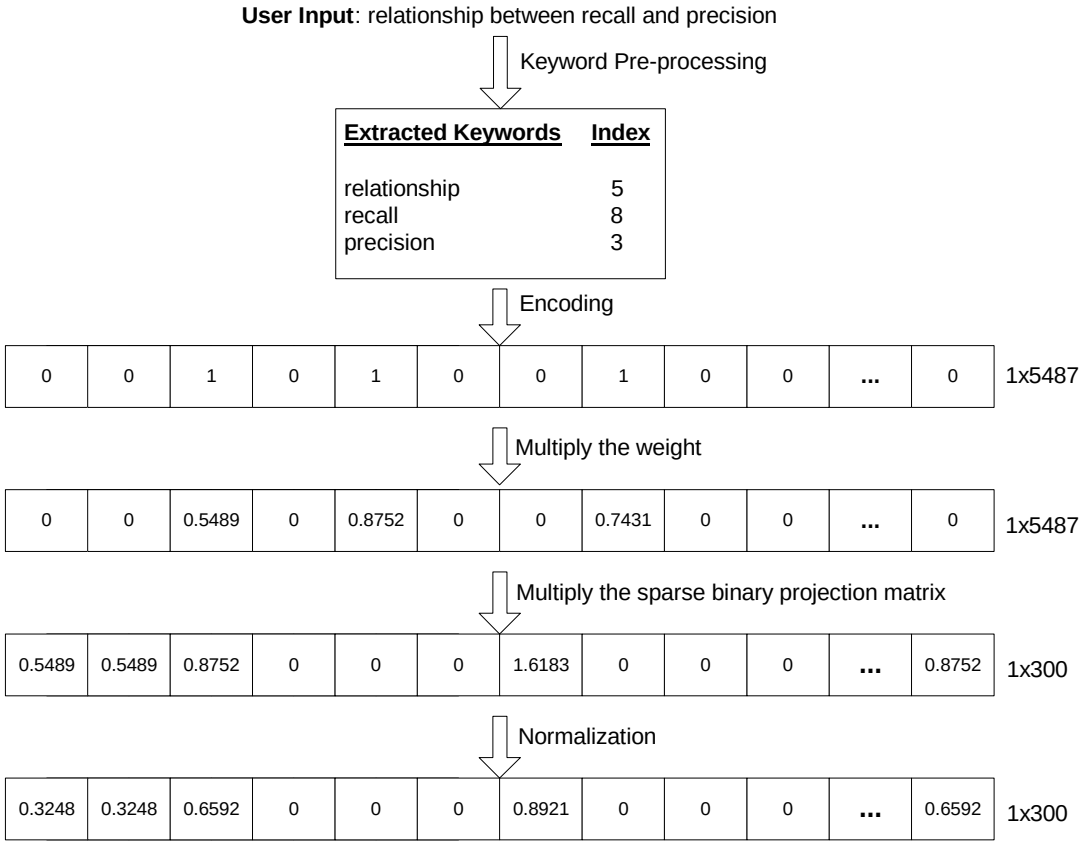


Figure 4-7. Example of encoding the user input keywords.

Figure 4-7 illustrates an example of encoding the user's input. User's input string is firstly tokenized and parsed into distinct keywords. As discussed in the previous section, a total of 5,487 distinct keywords are extracted from all the citations in the Web Citation Database. Thus, the user's input keywords are compared with these 5,487 words to form a 5,487-dimensional query vector. The occurrence of the keyword will set the element of the query vector in the corresponding position to 1. Otherwise, the keyword will be used as the index term to read from the WordNet thesaurus to find its synonyms. If any of the synonyms is found in the 5,487-keyword list, the original keyword in the query is replaced by the synonym. If it fails to find the matched keywords from the WordNet thesaurus, that element of the query vector will be set to 0.

During the weight multiplication step, instead of using the weight term TFIDF as used in the document vector, each element in the query vector is weighted using $qf \times idf$ (the frequency of the term in the query × the inverse document frequency of the term in the collection) (Turtle and Croft, 1991). This is used based on the assumptions that a content-bearing term that occurs frequently in the query is more likely to be important than one that occurs infrequently, and terms that occur infrequently in the document collection are more likely to be important than frequent or common terms.

After the above process, the query vector is multiplied by the same matrix R that is used in the training phrase to reduce its dimensionality to 300.

4.1.5.2 Document Retrieval Using Neural Network

This stage recalls the similar documents learned before and ranks them based on the closeness to the user's input query. The retrieval algorithm of the KSOM network is different from the Fuzzy ART network.

KSOM Network

In the KSOM neural network, the competitive learning is used to compute the winning cluster as in the training process. The Euclidean distance of the documents within the cluster to the user's input query is used to rank the documents. Figure 4-8 shows the cluster map in response to the query, "relationship between recall and precision". It includes the winning cluster and its neighbourhood clusters. The best-matched cluster is indicated to the user. In this case, the best matching cluster is 95. The interface also allows the user to browse through the cluster map. By clicking any of the cluster numbers in the cluster map, documents from that particular cluster are then listed and ranked according to the least Euclidean distance from the user's query.

This can be seen in Figure 4-9. The paper titles are underlined to allow users to get the full-text content of the paper through the underlying URL links. There are also "citing" and "cited" links provided, which allow the user to go deeper into the citing or cited documents of that particular publication.

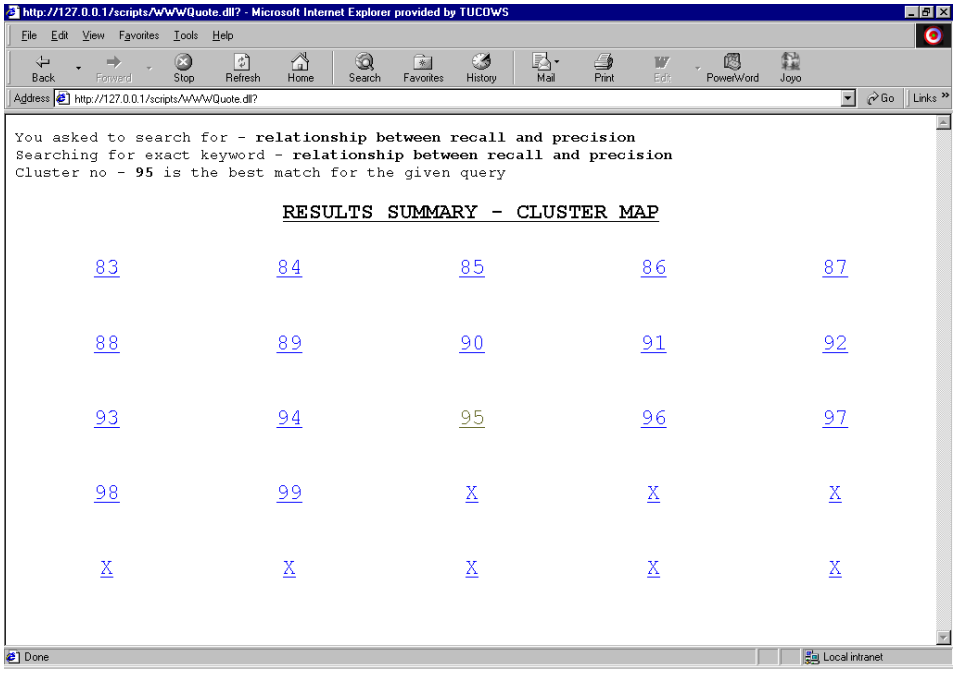


Figure 4-8. Cluster map for the KSOM algorithm.

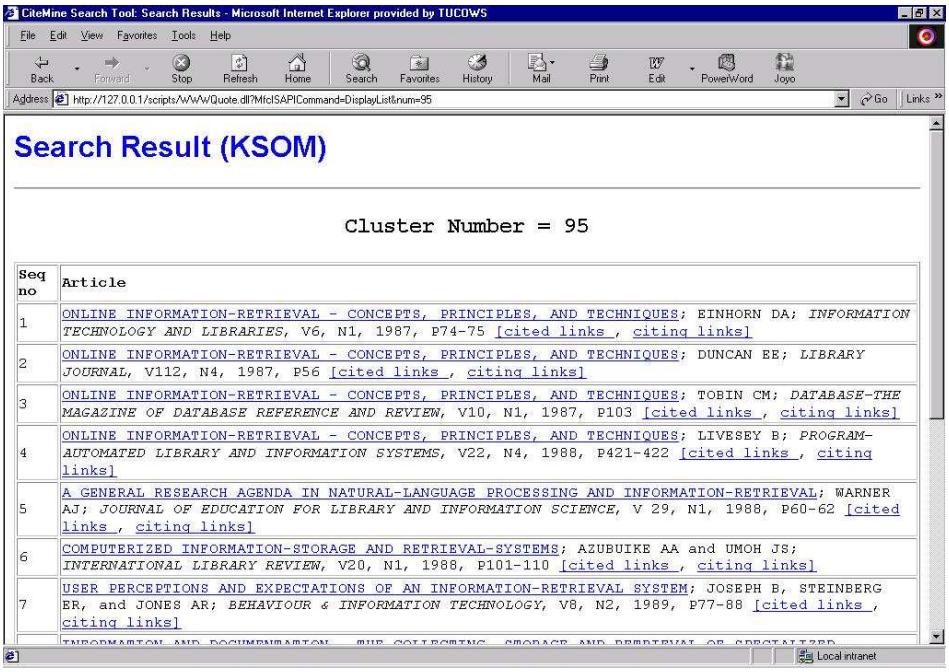


Figure 4-9. The result for the cluster number 95 using the KSOM algorithm.

Fuzzy ART Network

The recall of the stored documents in the Fuzzy ART neural network can be interpreted as identical to training. However, the vigilance test is always passed and no weight updates are performed during retrieval.

Figure 4-10 shows the search result for the same query "relationship between recall and precision" using the Fuzzy ART network. The total number of documents returned is 87, which is much greater than the total number of documents returned using the KSOM network, 32. Another difference from KSOM is that there are no neighborhood clusters information displayed using the Fuzzy ART network. This is because the Fuzzy ART network is sensitive to the sequence of the training data. In another words, the training data presented to the Fuzzy ART network in different sequence will result in different clusters to be generated. Therefore, there is no direct relationship between the winning cluster and its neighborhood clusters.

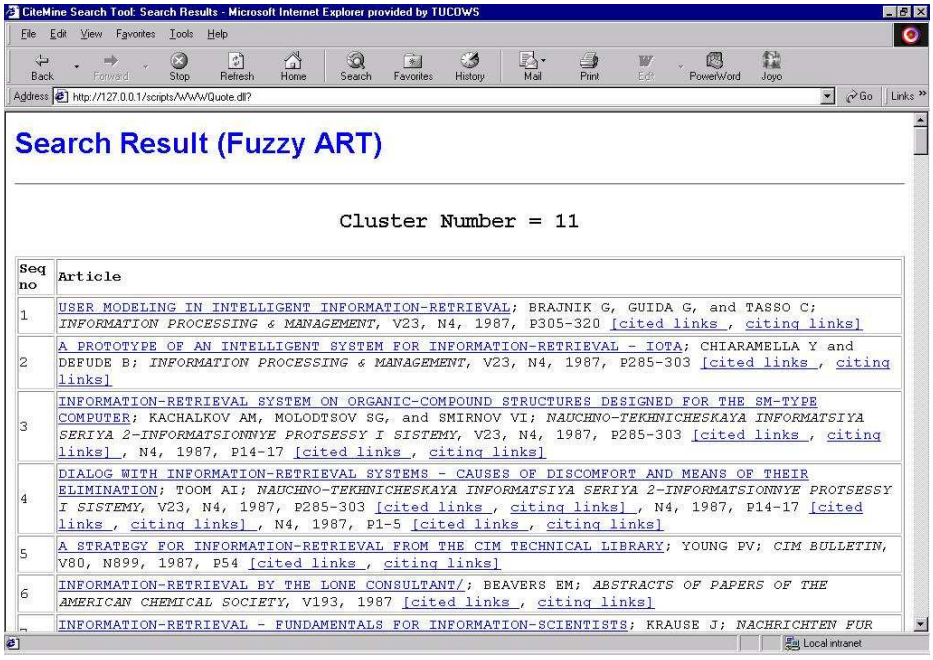


Figure 4-10. Search result of the Fuzzy ART network.

4.1.5.3 Document Ranking

The ranking of documents is done according to the Euclidean distance of the query term from all the documents in the cluster (Salton, 1991). Given a document vector *d* and a query vector *q*, their similarity $sim(d,q)$ is computed as follows:

$$sim(d,q) = \frac{\sum_{i=1}^{n}(w_{di} \times w_{qi})}{\sqrt{\sum_{i=1}^{n}(w_{di})^2 \times \sum_{i=1}^{n}(w_{qi})^2}}$$

where w_{di} and w_{qi} are the weight of the ith element in the document vector *d* and query vector *q* respectively. The formula actually measures the Euclidean distance between the document vector and the query vector. Any documents that have the smallest value is given the highest rank and the results in the cluster are sorted based on this principle. For the case where the values are the same, the ranking between the documents is done randomly.

4.2 Performance Evaluation

The performance evaluation has been conducted for both KSOM and Fuzzy ART neural networks. The training performance as well as the retrieval performance of these two models is compared. The experiments were carried out on a Pentium II 450 MHz machine with 128M RAM under the Windows NT operation system.

4.2.1 Training Performance

The performance of KSOM neural network's training depends on the initial weights and the selection of training parameters. The number of iterations should be expected to be reasonably large, as the learning is a statistical process. For good statistical accuracy, the number of steps should be at least 500 times larger than the

number of neurones (Kohonen, 1990). Here, the number of neurones was chosen to be 10×10 and the number of iterations should be at least 50,000. But with such large number of training steps, the algorithm is very computational and the intermediate results may be repeated when the training sets are large. In this research, the number of iterations was set to 5,000.

The vigilance threshold of the Fuzzy ART network determines the number of clusters generated. A value close to 1 indicates that a close match is required and there will be more clusters to be generated, while for smaller vigilance threshold, a poorer match is acceptable. For comparison purpose, the number of clusters generated by the Fuzzy ART network should be as close as possible to that of the KSOM network. Therefore, the vigilance threshold was set to 0.7 as the number of clusters generated is 129 which is quite close to the number of clusters generated by KSOM, i.e. 100.

The training performance is evaluated in two aspects, one is the training efficiency and the other is the training accuracy.

4.2.1.1 Training Efficiency

Table 4-2. Statistics on training efficiency of KSOM and Fuzzy ART.

Criteria	**KSOM**	**Fuzzy ART**
Pre-Processing Time	1 min 6 sec	1 min 6 sec
No. of Iterations	5,000	800
Training Time	46 min 25 sec	12 min 11 sec
Total No. of Clusters	100	129

The training efficiency can be measured based on the total training time and the number of iterations required by the neural networks to reach the convergent state. In this experiment, the following data was used. The number of keywords in the keyword list was 5,487. The number of words to be searched in the WordNet

dictionary was 121,962. The total number of document used for training (training set) was 1,000.

Table 4-2 shows the training performance of these two neural networks. It can be observed that Fuzzy ART requires much less training time as compared to KSOM. One of the reasons is that the number of iterations the Fuzzy ART network required to reach the convergent state is less than that in the KSOM network. The advantage for the Fuzzy ART network is that when there are new documents added into the document collection, only the additional new documents need to be presented to the Fuzzy ART network for training. Therefore, the training time will be much faster. But for KSOM, so long as the document collection changes, it needs to erase the previous memories and re-learn from the whole training samples again which makes it very time-consuming.

Another difference of these two neural networks is the total number of clusters generated. For KSOM, the number of clusters was pre-defined. It was set to be 100 in this experiment. For the Fuzzy ART model, the number of clusters generated depends on the vigilance threshold level. Here, the vigilance threshold was set to 0.7 to generate 129 clusters.

4.2.1.2 Training Accuracy

The training accuracy is measured by evaluating the effectiveness of cluster assignments. The standard recall, precision and F_1 measure are used here. Recall is defined as the ratio of correct assignments by the system divided by the total number of correct assignments. Precision is the ratio of correct assignments by the system divided by the total number of system's assignments. The F_1 measure, which was

introduced by Rijsbergen (Van Rijsbergen, 1979), combines recall (r) and precision (p) with an equal weight as follows:

$$F_1(r, p) = \left(\frac{2rp}{r + p} \right)$$

The F_1 scores can be calculated for each category and averaged across the experiments. Two kinds of averaging methods can be used, micro-averaging and macro-averaging techniques (Yang and Liu, 1999). Micro-averaging scores are computed on a per-document basis, they tend to be dominated by the system's performance on large categories. Macro-averaging scores are computed on a per-category basis, therefore, they are more likely to be influenced by the system's performance on small categories. Here, we only measure micro-averaging F_1 in the performance evaluation as it has been widely used in cross-method comparison (Yang and Liu, 1999). That is, the F_1 value is computed globally over all the $n \times m$ binary decisions, where n is the number of total training documents, and m is the number of categories. In addition, for comparison purpose, we also define *error* as the ratio of wrong assignments by the system divided by the total number of system's assignments.

Ten judges were selected to manually categorise documents into clusters. 1,000 documents were randomly selected for the sample measurement. The number of the manually categorized clusters was 20. These manually categorized clusters were used as the correct result with which the system's performance is compared. Therefore, the output of the KSOM network was also set to 20 as well. The vigilance threshold of the Fuzzy ART network was set to 0.715 in order to get the same number of clusters as in KSOM.

Table 4-3. Summary of training accuracy of KSOM and Fuzzy ART.

Algorithm	***Recall***	***Precision***	***F_1***	***Error***
KSOM	0.8143	0.8271	0.8206	0.0375
Fuzzy ART	0.7725	0.8007	0.7866	0.0486

Table 4-3 summarises the overall performance scores. It can be observed that KSOM performs better than Fuzzy ART on both recall and precision. Figure 4-11 shows more detailed performance scores for KSOM and Fuzzy ART. The horizontal axis is divided into equal-sized intervals for the training set frequency ranging from 50 to 1000. The vertical axis represents the F_1 score. The curves are obtained by averaging the per-category F_1 scores per interval for each neural network algorithm and interpolating the F_1 scores. Although at some points, Fuzzy ART performs better than KSOM, the overall performance of KSOM is still better than Fuzzy ART. It can also be observed that when the training set is large, the F_1 scores become stable.

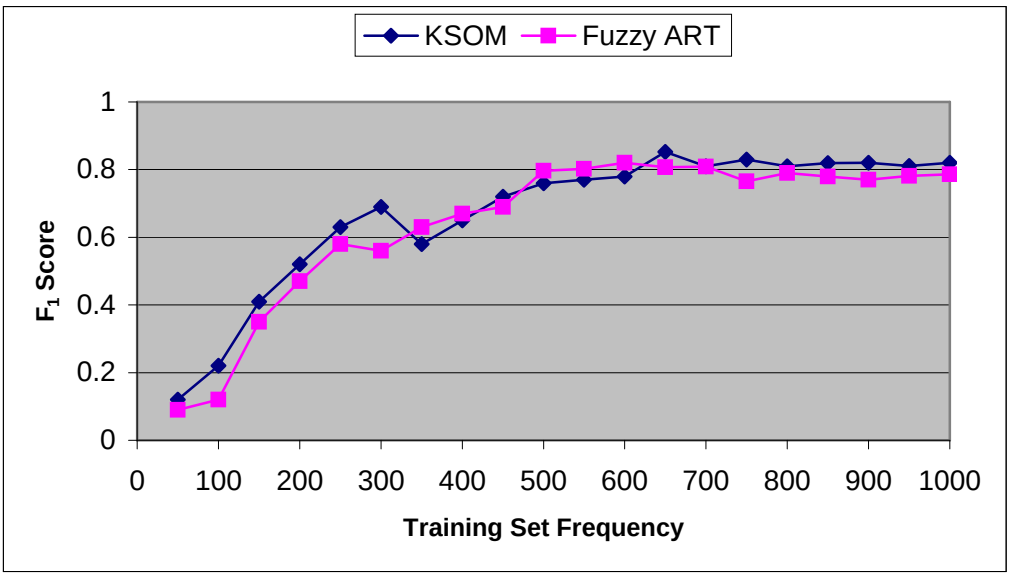


Figure 4-11. Performance of clustering accuracy for KSOM and Fuzzy ART.

4.2.2 Retrieval Performance

The retrieval performance is evaluated based on the average on-line retrieval speed, retrieval precision and recall. Retrieval speed measures how fast the result is presented to the user after submitting the search query. Precision is defined as the ratio of the number of documents that are judged as relevant for a particular query over the total number of documents retrieved. Recall is defined as the ratio of the number of relevant documents retrieved over the total number of relevant documents in the collection. Recall is considerably more difficult to measure than precision because it requires finding relevant documents that may not be retrieved in answering user's query (Blair and Maron, 1985). As the document corpus is quite large, it is difficult to get the total number of actual relevant documents according to the user's query. Therefore, the retrieval recall was not measured in this experiment.

As precision deals with the concept of relevance, we need to have a clear definition of the relevance. Harter (Harter, 1992) and Saracevic (Saracevic, 1996) classified the concepts of relevance into two main classes: objective or system-based relevance, and subjective or human (user)-based relevance. The objective relevance can be defined as a topically measure in the sense that a document is objectively relevant to a request if it deals with the topic of the request (Harter, 1992). The objective relevance is restricted to deal only with the degree to which the query representation matches the contents of the retrieved information objects. The subjective or user-based relevance is concerned with the aboutness and appropriateness of a retrieved information object and refers to the various degrees of intellectual interpretations carried out by human observers (Saracevic, 1995). This relevance measure is not based on the relationship between a query representation and

a retrieved information object, it is solely depends on the judgements of the users themselves.

In this research, we apply the above two kinds of relevance to evaluate the system. They are system-based relevance carried out directly at the objective processing level of the system by examining the ranked list of documents generated by the system, and user-based relevance associated with the user or user level.

There are many factors determining the retrieval precision, e.g. the accuracy of keywords pre-processing of training documents, the learning rate of the KSOM neural network, the vigilance threshold of the Fuzzy ART network, and the quality of user's input queries. If a user's input query contains many new keywords that are not found in the keyword list, the retrieval performance will be degraded. To conduct the experiment, 50 queries (listed in Appendix A) were formed with the keywords chosen carefully. The topics that closely related to the expected returned results for these queries were also clearly defined by the 10 persons such that these topics were used to evaluate the retrieval precision.

Search sessions were conducted to the system by the same 10 persons. Each person assessed the outcomes by perceiving whether the returned information object corresponds to the topical area required by the information need of the query. As the returned list might be very huge, it was not possible to examine every document one by one. Only the first 20 documents were assessed in each search session. The other documents were considered as irrelevant. For the returned list that contained less than 20 documents, all documents were examined. The assessment were made according to the three categories: highly relevant, partially relevant and not relevant (Pao, 1993). Three different values 1.0, 0.5 and 0.0 were assigned to the above categories

respectively. In addition, two more categories, 0.75 and 0.25, were added in order to make a finer assessment.

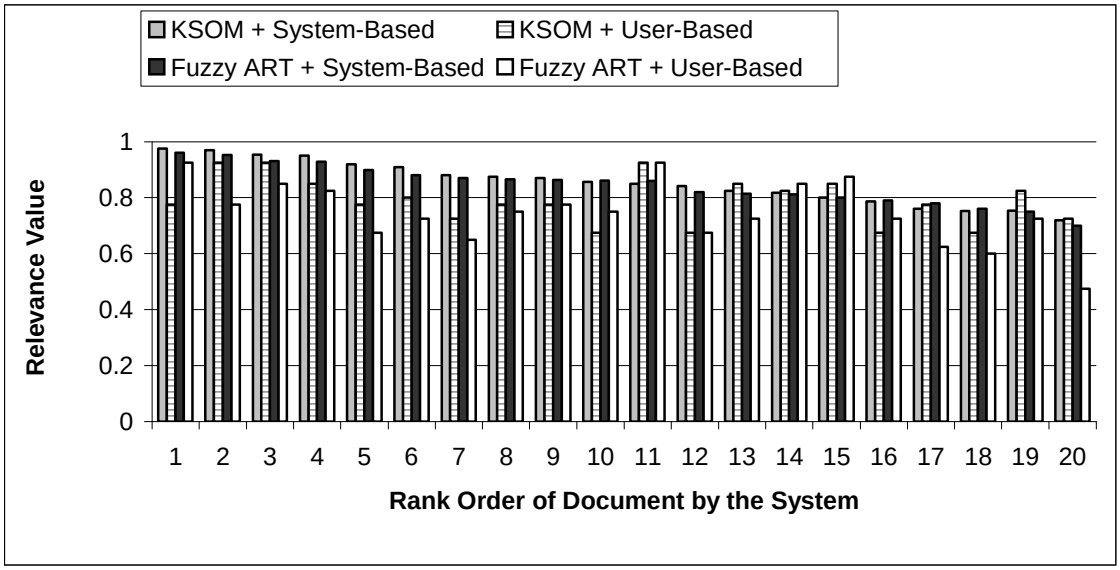


Figure 4-12. System-based versus user-based relevance for KSOM network.

The system-based relevance values and users' assessed relevance values for one search session using the KSOM and Fuzzy ART networks are compared in Figure 4-12. From the system's point of view, the algorithmic topically precision is quite high (85.45% for KSOM and 84.65% for Fuzzy ART). But from the test persons' perspectives, the precision is not so high as the overall average precision is 81.25% for KSOM and 74.5% for Fuzzy ART. The relevance measure for one query session is listed in Appendix B.

It can also be observed that the system-based relevance values are in descending order as the relevance values were calculated by the system itself. The user-based relevance values were calculated by summing up the results of all the test persons and averaged over these 10 persons. From the users' points of views, the relevance values of the first 20 documents are not in descending order. That is, documents that were listed behind may be more relevant than the documents appeared in the front.

The overall retrieval precision is obtained by averaging the system-based relevance and user-based relevance. The final result and the average retrieval speed are listed in Table 4-4. It can be observed that KSOM requires longer retrieval rate but with higher overall retrieval precision.

Table 4-4. Statistics on retrieval performance of KSOM and Fuzzy ART.

Technique	***Average Retrieval Speed***	***Retrieval Precision***		
		System-Based Relevance	**User-Based Relevance**	**Average**
KSOM	1.6 sec	84.35%	78.5%	81.43%
Fuzzy ART	0.7 sec	83.12%	71.25%	77.19%

In conclusion, both KSOM network and Fuzzy ART model are capable to uncover the semantic similarities of documents based on the feature vector representation of the documents. The results achieved using the KSOM network resemble the similarity of documents more faithfully, yet at the expense of longer training time. On the other hand, KSOM is not suitable for dynamic document database, as re-learning is required when there is a change in the document database.

4.3 Summary

In this chapter, document clustering using the KSOM and Fuzzy ART neural networks is discussed. It has been observed that the neural networks have a special way to store information. The vital information is not stored in memory locations. Rather, it is distributed throughout the neural network systems. The network architecture represents the knowledge and dynamically responds to the input. The ability to handle imprecise information makes the neural networks extremely fault tolerant and can give very good results which may not be achievable with any other techniques.

The performance evaluation for both KSOM and Fuzzy ART has been conducted. The results show that the system using KSOM achieves higher categorisation as well as retrieval precision. However, learning a new pattern in KSOM network, the existing or previous stored information in the network is affected. The Web Citation Database is dynamic in the sense that new citation information is kept on adding into the citation database whenever there are new scientific literature posted in the Web. Therefore, using KSOM to categorise and retrieve documents from the Web Citation Database is not suitable, as re-learning is required. On the contrary, the Fuzzy ART network exhibits a degree of plasticity when acquiring new training data. The recall of the data already learned is not affected, and the weights for clusters already stored are not modifiable. To conclude, both KSOM and Fuzzy ART networks have their advantages and disadvantages, but Fuzzy ART network algorithm is more suitable to be applied in the dynamic environment such as Web Citation Database.

Chapter 5
Data Mining for Author Clustering

Traditional study of the intellectual structure of the scholarly community is usually carried out by utilizing the author co-citation analysis. As discussed in Chapter 2, two authors are correlated if the frequency of their works that are cited together by another publication is very high. We can then predict that the research interest of these two authors must be similar or related. In this chapter, the data mining process for author clustering is discussed.

5.1 Data Mining Process

Figure 5-1 shows the data mining process for author clustering. It consists of the following six steps:

1. *Create Author Co-Citation Pairs*. The author co-citation pairs are created from the Web Citation Database by grouping two distinct authors together with the same "source_ID" in the CITATION table.
2. *Create Raw Co-Citation Matrix*. The co-citation frequency of each author pair is calculated first, which is then used to compute the co-citation link strength. The author pairs with the co-citation link strength below a certain threshold are filtered out and the rest will form the raw co-citation matrix.
3. *Convert into Correlation Matrix*. The raw co-citation matrix is converted into the correlation matrix by substituting the author co-citation count with the Pearson's correlation coefficient.

4. *Generate Author Clusters*. The Agglomerative Hierarchical Clustering (AHC) algorithm is applied to the correlation matrix to generate the author clusters.
5. *Display Author Cluster Map.* The Multidimensional Scaling (MDS) technique is used to generate XY-coordinates for all the authors in the two-dimensional space, which are then combined with the cluster knowledge to display author cluster maps.
6. *Author Retrieval*. The author cluster information is used to perform author cluster retrieval.

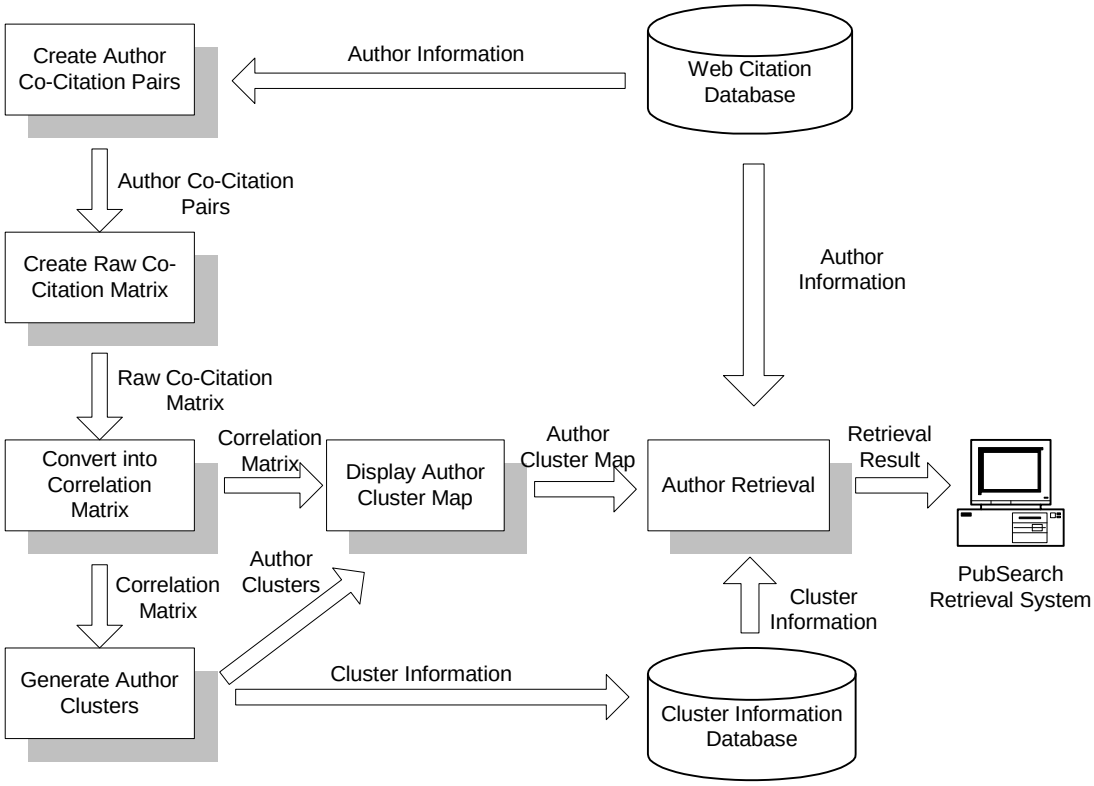


Figure 5-1. Data mining process for author clustering.

5.1.1 Create Author Co-Citation Pairs

The input to this step is the CITATION table from the Web Citation Database. One of the attributes of the CITATION table is “source_ID”, which specifies the source paper of the current record. The records with the same “source_ID” are from the same source paper. Therefore, the author co-citation pairs can be created by

grouping two distinct authors together with the same "source_ID". The co-citation count of each author pair is calculated as the co-citation frequency. Only the first author of any cited papers is taken into consideration. That is, authors who are published as co-authors in the second or later positions will not be used in this step.

Table 5-1. A part of the CITATION table.

citation_ID	***source_ID***	***Author1***	**…**
7	3	Boswell P	…
8	3	Finch C	…
9	3	Garner WR	…
10	3	Gray B	…
11	3	Levey M	…
13	3	Macgregor J	…
14	3	Macgregor J	…

Table 5-2. Author co-citation pairs.

Sequence No	***Author Co-Citation Pairs***	
1	Boswell P	Finch C
2	Boswell P	Garner WR
3	Boswell P	Gray B
4	Boswell P	Levey M
5	Boswell P	Macgregor J
6	Finch C	Garner WR
7	Finch C	Gray B
8	Finch C	Levey M
9	Finch C	Macgregor J
10	Garner WR	Gray B
11	Garner WR	Levey M
12	Garner WR	Macgregor J
13	Gray B	Levey M
14	Gray B	Macgregor J
15	Levey M	Macgregor J

For example, Table 5-1 displays a part of the CITATION table. All the citation records were extracted from the same source paper according to the same "source_ID". A total of 15 author co-citation pairs can be generated as shown in Table 5-2. It should be noted that although the author "Macgregor J" appears twice in the citation records with the "CITAION_ID" equals to "13" and "14" respectively, it will only be treated as one author in the creation of author co-citation pairs. In another words, the author pair, Macgregor J-Macgregor J, will not be created.

5.1.2 Create Raw Co-Citation Matrix

After the author co-citation pairs are created, the co-citation link strength (Garfield, 1980) is calculated using the following formula:

$$\text{Link Strength (AB)} = X / (Y - X)$$

where X is the number of co-citations of author A and author B, Y is the sum of total number of citations of A and total number of citations of B. In fact, this formula normalizes the co-citation link strength by taking into the account of the total number of citations for both A and B.

A threshold is set to filter out the author pairs with insignificant co-occurrences. Author pairs with co-citation link strength exceeding the threshold value are retained. The raw co-citation matrix can then be formed by taking the list of authors as the entries in both row and column. The value of each cell in the matrix is the co-citation frequency count of the authors in the corresponding row and column. Such matrix is symmetric, as the lower triangular matrix is identical to the upper triangular matrix.

For each diagonal cell of the matrix, White and Griffith (White and Griffith, 1981) select the value based on the highest off-diagonal co-citation counts for each

author. Chen and Carr (Chen and Carr, 1999) put the mean co-citation counts of the same author in the diagonal cell. As the diagonal cells store the self-citation counts, they do not contribute to the author clustering process. Here, the diagonal cell values are treated as zeros. An example of the raw co-citation matrix with co-citation frequency values is shown in Table 5-3.

Table 5-3. An example of a 10×10 raw co-citation matrix with co-citation frequency.

AUTHOR	Anderson	Baddley	Bates	Belkin	Blair	Bookstain	Borgman	Brainerd	Cleverdon	Cooper
Anderson	0	53	9	10	3	2	12	45	1	0
Baddeley	53	0	0	0	0	0	0	89	0	0
Bates	9	0	0	287	74	27	187	0	24	29
Belkin	10	0	287	0	91	67	166	0	141	89
Blair	3	0	74	91	0	32	38	0	34	55
Bookstain	2	0	27	67	32	0	13	0	14	136
Borgman	12	0	187	166	38	13	0	0	25	13
Brainerd	45	89	0	0	0	0	0	0	0	0
Cleverdon	1	0	24	141	34	14	25	0	0	52
Cooper	0	0	29	89	55	136	13	0	52	0

The link strength of each author co-citation pair is calculated and the result is shown in Table 5-4. For example, if the link strength threshold is set to 0.13, for the author co-citation pairs containing “Anderson”, only Anderson-Baddley and Anderson-Brainerd pairs will be retained.

Table 5-4. An example of a 10×10 raw co-citation matrix with link strength.

AUTHOR	Anderson	Baddley	Bates	Belkin	Blair	Bookstain	Borgman	Brainerd	Cleverdon	Cooper
Anderson	0	0.2849	0.0259	0.0210	0.0098	0.0068	0.0338	0.2239	0.0042	0
Baddeley	0.2849	0	0	0	0	0	0	1.5345	0	0
Bates	0.0259	0	0	1.3165	0.2913	0.0941	0.9444	0	0.1026	0.0986
Belkin	0.0210	0	1.3165	0	0.2473	0.1772	0.4743	0	0.5685	0.2438
Blair	0.0098	0	0.2913	0.2473	0	0.1356	0.1262	0	0.1910	0.2477
Bookstain	0.0068	0	0.0941	0.1772	0.1356	0	0.4761	0	0.0761	1.0709
Borgman	0.0338	0	0.9444	0.4743	0.1262	0.4761	0	0	0.1025	0.0405
Brainerd	0.2239	1.5345	0	0	0	0	0	0	0	0
Cleverdon	0.0042	0	0.1026	0.5685	0.1910	0.0761	0.1025	0	0	0.3355
Cooper	0	0	0.0986	0.2438	0.2477	1.0709	0.0405	0	0.3355	0

The value of the link strength threshold affects the final results of the author clustering process as the greater the frequency of a given pair, the greater the

likelihood that the two authors belong to the same research area. Therefore, the scope of the research area can be adjusted by increasing or decreasing the threshold. Generally, the higher the threshold, the narrower the scope. In this research, the link strength threshold is set to 0.45 which is determined experimentally.

5.1.3 Convert into Correlation Matrix

If two authors are cited frequently with a third author separately, but infrequently with others, then they will have a high positive correlation. This can be perceived as related or similar by the citing population. That is, even though the two authors may not be highly co-cited, they may still present certain degree of similar co-citation profile. For example, author A is highly cited with author C but not with other authors. Similarly, author B is also highly cited with author C but not with other authors. The overall co-citation count for authors A and B may be small. But they still exhibit the similar co-citation profiles as both authors are cited frequently with author C. The creation of correlation matrix can help to remove the differences between authors who are highly cited and those who have similar profiles but are less frequently cited overall (Kerlinger, 1973).

The correlation is defined as a measure of similarity. The higher the positive correlation, the more similar the two authors are from the perceptions of citers. The Pearson's correlation coefficient is normally used in author co-citation analysis to measure the similarity between author pairs (White and McCain, 1998). The formula used to calculate the Pearson's correlation coefficient ***r*** (Johnson, 1988) is given as follows:

$$r = \frac{N(\sum XY) - (\sum X)(\sum Y)}{\sqrt{[N(\sum X^2) - (\sum X)^2][N(\sum Y^2) - (\sum Y)^2]}}$$

where X and Y are two input vectors and N is the dimension of the input vector. For example, if a raw co-citation matrix of 40 authors is generated, for an author co-citation pair A and B, each element of input vector X will be the co-citation count of author A with all the other authors, each element of input vector Y will be the co-citation count of author B with all the other authors, and N will be 39 as we do not take the self-citation count into consideration.

The Pearson's correlation coefficient r measures the strength of linear correlation. It is always between –1 and +1 inclusive. –1 means perfect negative linear correlation and +1 means perfect positive linear correlation. This correlation coefficient r does not change if the X and Y variables are interchanged. An example of the correlation matrix is shown in Table 5-5:

Table 5-5. An example of a 10×10 correlation matrix.

AUTHOR	Anderson	Baddley	Bates	Belkin	Blair	Bookstain	Borgman	Brainerd	Cleverdon	Cooper
Anderson	0	0.7813	0.1914	0.1865	0.1807	0.12	0.1562	0.6251	0.1344	0.1209
Baddeley	0.7813	0	0.018	0.0125	0.0166	0.0024	0.0127	0.5939	0.0046	0.0001
Bates	0.1914	0.018	0	0.9288	0.8013	0.5206	0.9289	0.0146	0.6883	0.6123
Belkin	0.1865	0.0125	0.9288	0	0.8554	0.6757	0.8274	0.0107	0.8438	0.786
Blair	0.1807	0.0166	0.8013	0.8554	0	0.7175	0.7191	0.0122	0.7845	0.7933
Bookstain	0.12	0.0024	0.5206	0.6757	0.7175	0	0.452	0.0021	0.6553	0.8931
Borgman	0.1562	0.0127	0.9289	0.8274	0.7191	0.452	0	0.0049	0.6063	0.5276
Brainerd	0.6251	0.5939	0.0146	0.0107	0.0122	0.0021	0.0049	0	0.0041	0
Cleverdon	0.1344	0.0046	0.6883	0.8438	0.7845	0.6553	0.6063	0.0041	0	0.8366
Cooper	0.1209	0.0001	0.6123	0.786	0.7933	0.8931	0.5276	0	0.8366	0

From the co-citation pairs containing the author "Anderson", it can be easily found that the author pair, Anderson-Baddeley, is the most highly co-cited one. Therefore, these two authors may belong to the same research area.

5.1.4 Generate Author Clusters

The correlation matrix stores the inter-author proximity values. It serves as the input to the Agglomerative Hierarchical Clustering (AHC) algorithm to derive the author clusters. Figure 5-2 shows the basic process of the AHC algorithm.

AHC_Algorithm:

Input: A set of N items (or authors) to be clustered, and an N×N correlation matrix.

Process:

1. Start by assigning each item to its own cluster. That is, N clusters will be created initially.
2. Find the most similar pair of clusters and merge them into a single cluster.
3. Compute the similarities between the new cluster and each of the old clusters. There are four different methods to compute the similarities, namely, the single link, complete link, average link, and Ward's method.
4. Repeat steps 2 and 3 until the desired number of clusters is reached.

Output: The desired number of clusters.

Figure 5-2. The AHC algorithm to generate author clusters.

As shown in Figure 5-2, the similarity measure in step 3 can be done in different ways, using the single link, complete link, average link clustering, and Ward's method as follows.

- *Single link or nearest neighbor method*. In this method, the similarity between one cluster and another cluster equals to the greatest similarity from any members of one cluster to any members of the other clusters. For clusters C_A and C_B, the similarity S_{AB} between them is defined as:

$$S_{AB} = \min(S(N_A, N_B))$$

  where $N_A \in C_A$ and $N_B \in C_B$ and $S(N_A, N_B)$ is the similarity value between N_A and N_B.

- *Complete link or furthest neighbor method.* This method is the opposite of the single link method, the least similarity pair between two clusters defines the inter-cluster similarity. For clusters C_A and C_B, the similarity S_{AB} between them is defined as:

$$S_{AB} = \max(S(N_A, N_B))$$

where $N_A \in C_A$ and $N_B \in C_B$ and $S(N_A, N_B)$ is the similarity value between N_A and N_B.

- *Average link method.* The average link clustering uses the average pair-wise similarity between two clusters as the inter-cluster similarity. For clusters C_A and C_B, the similarity S_{AB} between them is defined as:

$$S_{AB} = avg(S(N_A, N_B))$$

where $N_A \in C_A$ and $N_B \in C_B$ and $S(N_A, N_B)$ is the similarity value between N_A and N_B.

- *Ward's method*. The loss of information that results from the grouping of individuals into clusters can be measured by the total sum of squared deviations of every point from the mean of the cluster to which it belongs. This method joins two clusters that result in the smallest increase in the overall sum of the squared intra-cluster distance. For clusters C_A and C_B, the Error Sum of Square (ESS) is computed as:

$$ESS = \sum_{i=1}^{n} X_i^2 - \frac{1}{n}(\sum_{i=1}^{n} X_i)^2$$

where X_i is the element in C_A or C_B, n is the total number of elements in C_A and C_B. The *ESS* is calculated for every pair of clusters. The two clusters with the smallest *ESS* are joined together to become one cluster.

In the implementation, linked list is used to store the similarity matrix data. If the matrix is N×N, there are N elements in the main linked list. For each element, there is another linked list to store the similarity value between this element with all the other elements. Initially, there are N clusters with each cluster contains a single element. The pair with the greatest similarity is found. Then, these two elements are deleted from their corresponding entries and a new entry is created for these two

elements. The similarities between this cluster with all the other elements are then calculated. This can be considered as a merge process. A stopping threshold is defined such that no clusters can be merged if all the similarity values are below this threshold.

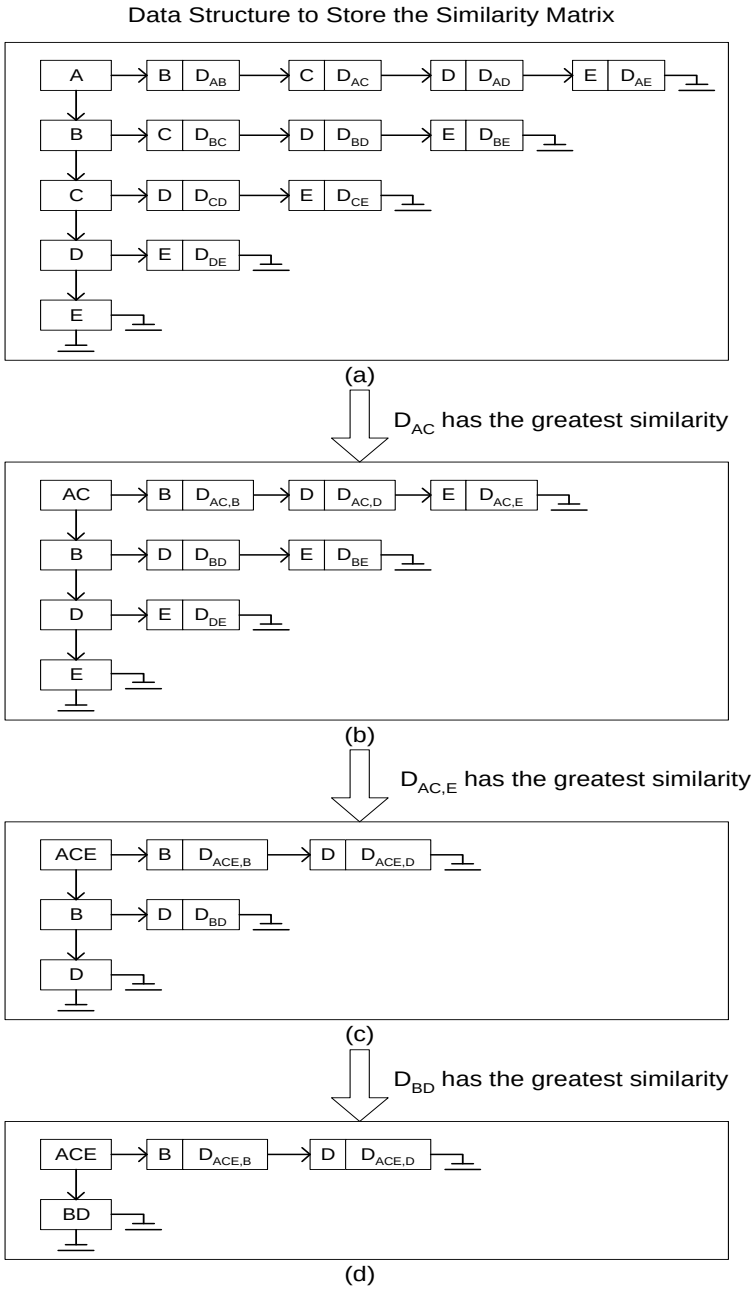


Figure 5-3. Data representations of the similarity matrix and clustering result.

Figure 5-3 shows the data representations of the similarity matrix and cluster result. The input similarity matrix is 5×5, which contains five authors, author A, author B, author C, author D, and author E. Initially, these five authors are stored in a

linked list as shown in Figure 5-3 (a). Each entry of this linked list stores another linked list that contains the similarity value with all the other authors. For example, for the slot of author A, it contains another linked list that stores the similarity values between author A and authors B, C, D, and E respectively, which are D_{AB}, D_{AC}, D_{AD}, and D_{AE}. At the beginning, there are five clusters with each cluster only containing one author.

After the first scanning of the linked list stored in the slot for author A, the similarity value between A and C (D_{AC}) is found to be the greatest among the values between A and all the other authors and it is greater than the stopping threshold. Therefore, the slot for author A is now occupied by the author pair A and C, which is treated as a single element. The similarity values between the author pair AC with all the other authors B, D, E are re-calculated and stored in the linked list as shown in Figure 5-3 (b). On the other hand, authors A and C are grouped into one cluster, therefore, the main linked list only contains four entries now with authors A and C stored in the same slot. As A and C are treated as a single element now, all the other entries which contain the similarity values between C and other elements are deleted. For example, the entry containing the similarity value between B and C is deleted. This can be observed in Figure 5-3 (b) where the linked list for author B only has two entries that contain the similarity values between B and authors D and E respectively. This process repeats until no similarity values are greater than the stopping threshold. In this example, two clusters are finally obtained, with A, C, and E are grouped into one cluster while B and D are grouped into another cluster as shown in Figure 5-3 (d).

Deciding the number of clusters is essential for hierarchical clustering. The clustering method begins by considering each author to be a cluster. At each stage of the analysis, the algorithm combines two clusters until, at the end, all of the authors are

in a single cluster. We need to define the stopping rule such that at certain point, the algorithm will quit from the cluster merge process and return the clusters at that stage.

As the hierarchical clustering approach used is agglomerative, it is obvious that the first two clusters merged are the ones that show the greatest similarity. However, when the number of authors per cluster increases by merging the existing clusters, the authors within each cluster are increasingly dissimilar as the process goes on. In this step, the retrieval efficiency also needs to be considered, and each resulting cluster should contain a reasonable number of authors. To satisfy the above considerations, the threshold for the stopping rule is set to 0.5 such that two clusters will only be merged when the inter-cluster similarity is greater than the threshold (Zamir and Etzioni, 1998).

5.1.5 Display Author Cluster Map

After the clusters are generated, it is necessary to present the author cluster information in a graphical manner to make it more effective and intuitive. Traditionally, ACA research relies on the Multidimensional Scaling (MDS) program in the SPSS statistical package to generate the author cluster map. It requires human interpretation to identify author clusters. Chen (Chen, 1999; Chen and Carr, 1999) incorporates visualization techniques into citation analysis to generate the author cluster map in three-dimensional world. The citation patterns are extracted from document collections using Latent Semantic Indexing (Deerwester *et al.*, 1990) and Pathfinder Network Scaling (Schvaneveldt *et al.*, 1989).

Modified MDS Algorithm:

Input: An N × N correlation matrix (D).

Process:

1. Assign author points to coordinates in a two-dimensional space arbitrarily.
2. Compare Euclidean distances among all the pairs of points to form a matrix, which is called a Dhat matrix.
3. Compare the Dhat matrix with the original correlation matrix D by evaluating the *stress* (Chatfield and Collins, 1989) function. The smaller the *stress* value, the greater the correspondence between them. The general form of the *stress* function is given as:

$$stress = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(f(x_{ij}) - d_{ij})^2}{scale}}$$

where N is the original dimension of the similarity matrix;

d_{ij} refers to the Euclidean distance between points i and j on the two-dimensional map;

$f(x_{ij})$ is the transformation function of the original input data of the similarity matrix; in metric scaling, $f(x_{ij}) = x_{ij}$; and in non-metric scaling, $f(x_{ij})$ is a weak monotonic transformation of the input data that maximizes the stress function;

scale refers to a constant scaling factor, it is used to keep the *stress* value between 0 and 1.

When the MDS map reproduces the input data perfectly, the *stress* is zero. Thus, the smaller the *stress*, the better the representation.

4. Adjust the coordinates of each point in the direction that can best minimize the *stress*.
5. Repeat step 2 through step 4 until the *stress* will not get any lower.

Output: XY-coordinates for each author.

Figure 5-4. The modified MDS algorithm to generate author cluster maps.

Here, we propose another method to generate the author cluster map. It uses MDS algorithm (Green *et al.*, 1989) to map authors in the correlation matrix into two-dimensional space with each point representing one author. The modified MDS algorithm is given in Figure 5-4. The output of the algorithm is the XY-coordinate of each author. Then, based on the author cluster information and XY-coordinates of all the authors, a two-dimensional map is generated as shown in Figure 5-5. Authors from various clusters are differentiated using points with different shapes and colors. Each cluster represents a research area. Authors within the same cluster are the experts or researchers of the same research area.

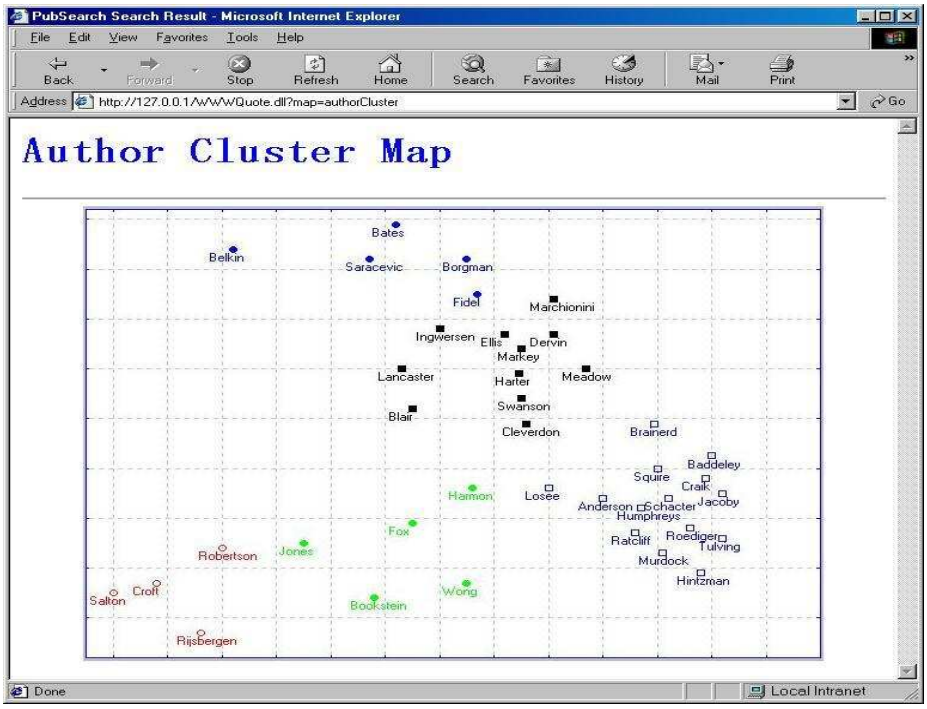


Figure 5-5. Author cluster map.

5.1.6 Author Retrieval

The author cluster information and cluster map have been incorporated into the author retrieval process of the PubSearch retrieval system. The user's query input will be author name, either full name or partial name. The system will then display the map of the author cluster that the searched author belongs to.

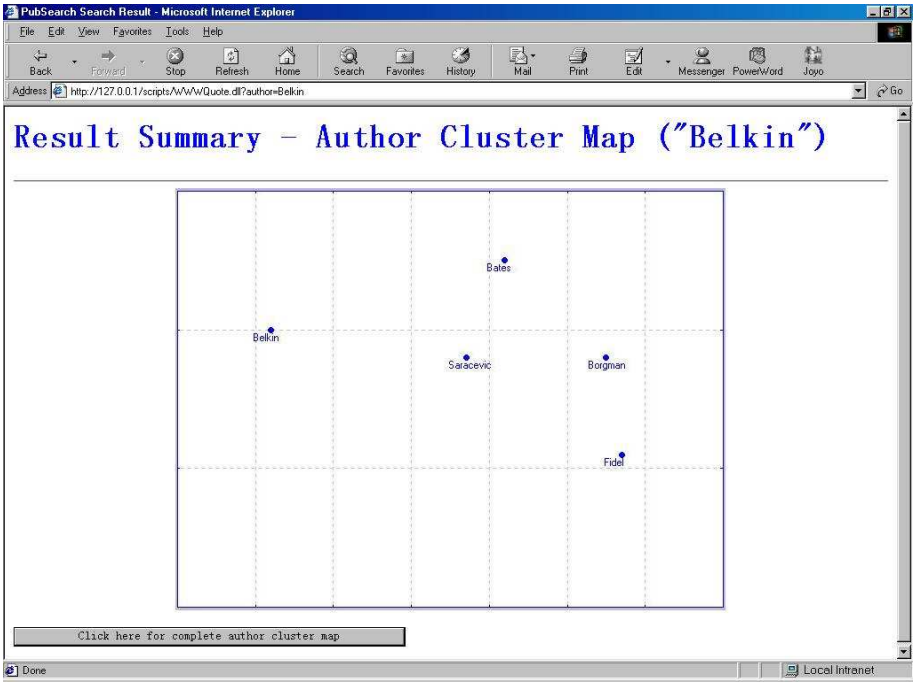


Figure 5-6. Author cluster map for the search query on "Belkin".

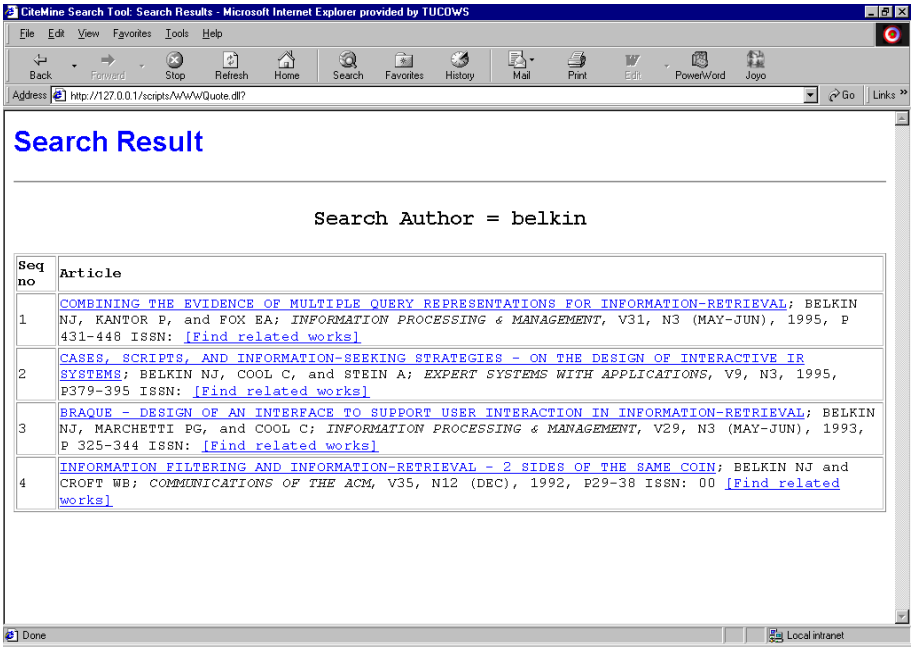


Figure 5-7. List of papers by "Belkin".

For example, if the user's query is "Belkin", Figure 5-6 shows the map of the cluster that contains the author "Belkin". Each point represents an author. The distance between each other roughly corresponds to the similarity among them. By clicking any author names, a list of papers written by that author are displayed as shown in Figure

5-7. All the paper titles are underlined to indicate the availability of the URL links for full-text access of the publications. Users are also allowed to view the author cluster map by clicking the button “Click here for overall author cluster map” in Figure 5-6. The author cluster map will then be displayed as shown in Figure 5-5.

Besides author information retrieval, the author cluster maps can also be used to monitor the change of the research focus of a particular author as well as the newly emerged research areas throughout the years. This is illustrated as follows. Forty most highly cited authors from the test citation database in information retrieval field from 1987 to 1997 are used as author samples. Users are allowed to enter the range of the publication year such that author cluster maps are generated in different time frames. For example, the 10-year period is divided into two time frames, 1987 to 1991 and 1992 to 1997. The records with the publication date falling within these two time frames are analyzed separately and two author cluster maps are formed as shown in Figure 5-8 and Figure 5-9 respectively.

For illustration purposes, the boundaries of every author cluster are manually drawn and the description for each research area is given based on the common research topics of every author within a cluster. Authors from different research areas, such as the general IR theory, IR techniques, IR model, user information seeking and retrieving behavior, are displayed. The area of computerized IR system & mathematical model during the period of 1987-1991 is more or less similar to the area of IR system design and evaluation during the period of 1992-1997. The newly emerged research fields include user perspectives of IR and IR theory research. Some authors’ research interests have changed over these two periods. For example, Belkin shared the same research interest with Van Rijsbergen, Croft and Sparck Jone during

the period of 1987-1991. But his research focus subsequently shifted to user information seeking and retrieving behavior.

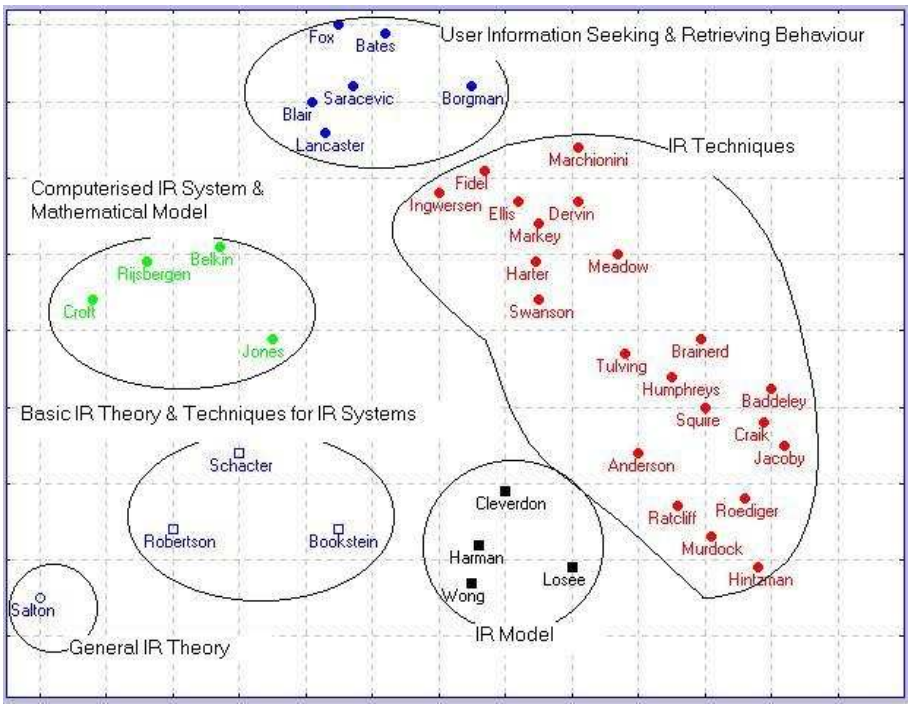


Figure 5-8. Author cluster map (1987-1991).

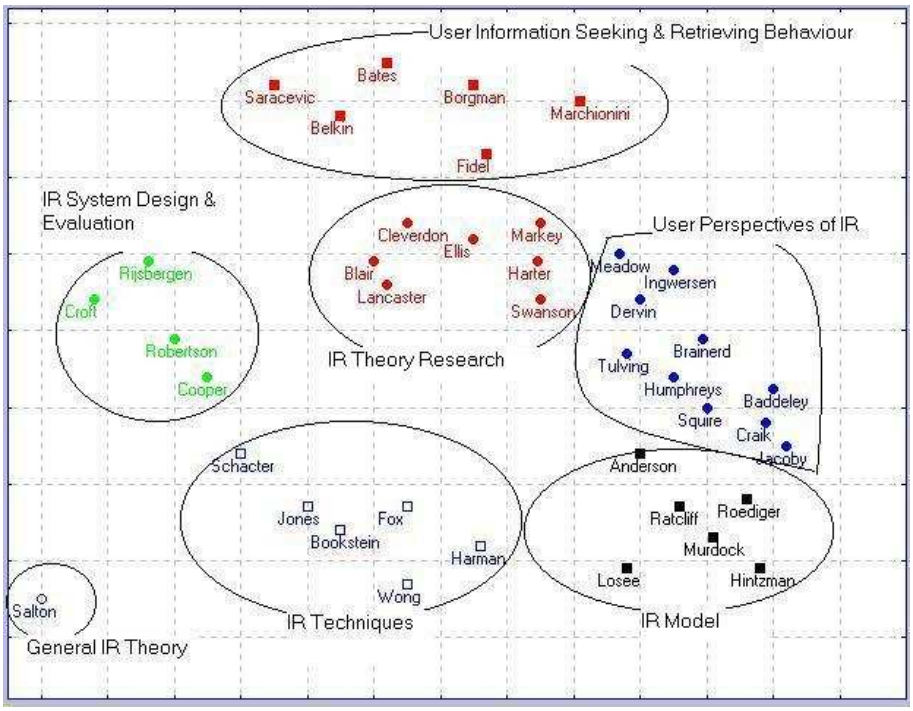


Figure 5-9. Author cluster map (1992-1997).

5.2 Performance Analysis

The performance evaluation has been conducted on the clustering results. The entropy measure that uses labels as defined in (Boley, 1998) is applied here. Each author is manually given a label according to the research area the author belongs to. The labels are used to measure the entropy of the resulting cluster as a measure of quality. The entropy of a given cluster *C* is defined by the following formula:

$$e_C = -\sum_i \left(\frac{c(i,C)}{\sum_i c(i,C)}\right) \log\left(\frac{c(i,C)}{\sum_i c(i,C)}\right)$$

where $c(i,C)$ is the number of times label i occurs in cluster *C*. The entropy for a cluster is zero if the labels of all the authors are the same, that is, all the authors are in the same research area. Otherwise, it is positive. The total entropy is the weighted average of the individual cluster's entropies:

$$e_{total} = \frac{1}{M}\sum_C (e_C \times N_C)$$

where M is the total number of clusters and N_C is the number of authors in cluster *C*. Therefore, the lower the entropy, the better the quality of the author clustering algorithm.

In the proposed author clustering process, two factors affect the final clustering results. The first one is the co-citation link strength threshold. As recalled from the previous sections, only author pairs with the link strength greater than the pre-defined threshold are used to form the correlation matrix. Other author pairs are discarded from the measurement. Experiments need to be conducted in order to get the optimal co-citation link strength threshold. Another factor that can affect the clustering results is the method used to calculate the similarity between clusters. Four different methods

were implemented to compare the performance. They were the single link, complete link, average link and Ward's method.

5.2.1 Experiment

The whole test citation database consisted of papers on information retrieval studies published from 1987 to 1997, which was very large. For experimental purpose, we had chosen 40 most highly cited authors as the input data to do the experiment. These authors had been classified into six different research areas (Ding, 1998) using the SPSS tool with manual processing. Table 5-6 shows the classification result.

Table 5-6. The classified author clustering results.

Research Area	***Author Names***
General IR theory	Salton G
Computerised IR system & mathematical model	Croft WB, Jones KS, Robertson SE, Vanrijsbergen CJ
IR model	Bookstein A, Cooper WS, Fox EA, Harman D, Losee RM, Wong SKM
Psychology	Anderson JR, Baddeley AD, Brainerd CJ, Craik FIM, Hintzman DL, Humphreys MS, Jacoby LL, Murdock BB, Ratcliff R, Roediger HL, Schacter DL, Squire LR, Tulving E
IR theory & techniques (less mathematical & computerized)	Blair DC, Cleverdon CW, Dervin B, Ellis D, Harter SP, Ingwersen P, Lancaster FW, Marchionini G, Markey K, Meadow CT, Swanson DR
User searching behaviour	Bates MJ, Belkin NJ, Borgman CL, Fidel R, Saracevic T

5.2.2 Co-Citation Link Strength Threshold

In this section, experiments were conducted to show how the co-citation link strength affects the final clustering result. The complete link method was used for similarity measure as it can achieve the best clustering result that will be shown later. Table 5-7 shows how documents were distributed to four different clusters by research area labels when co-citation link strength was set to 0.3. The entropy (e_C) for each cluster was also calculated.

Table 5-7. Statistics of entropy for each cluster with co-citation link strength = 0.3.

Research Area	***Clusters***			
	1	***2***	***3***	***4***
General IR theory	0	0	1	0
Computerised IR system & mathematical model	1	3	0	0
IR model	0	0	6	0
Psychology	9	1	0	3
IR theory & techniques (less mathematical & computerized)	0	1	1	9
User searching behaviour	0	0	0	5
Entropy (e_C)	0.2611	0.2741	0.0947	0.2183
Total entropy (e_{total})				2.1126

By varying the co-citation link strength threshold, different clustering results are obtained. Figure 5-10 shows a comparison of the entropy values calculated by varying the co-citation link strength threshold from 0.3 to 0.7 using the complete link method. The optimal value for the co-citation link strength threshold is 0.45, and the corresponding entropy value obtained is 0.8562, which gives the best clustering result. It should be noted that increasing or decreasing the threshold results in poor accuracy.

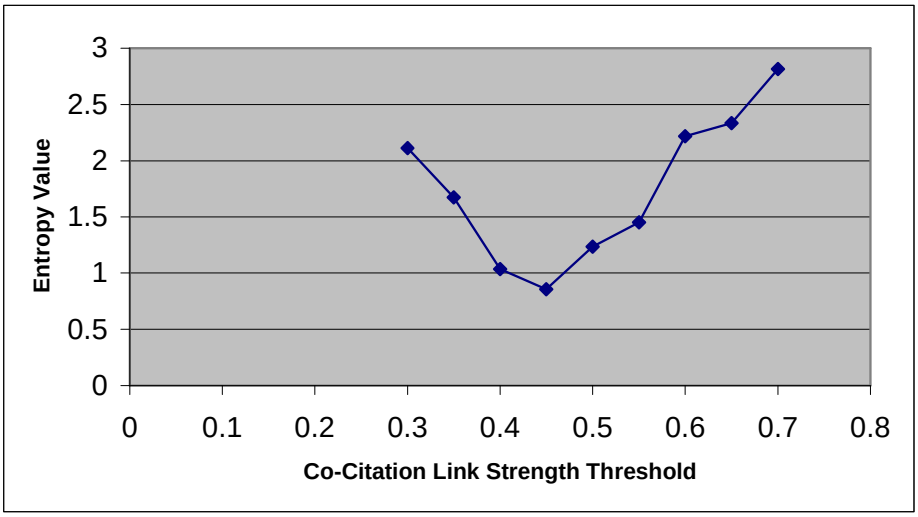


Figure 5-10. Entropy values by varying the co-citation link strength threshold.

5.2.3 Comparison of Different Similarity Measure Methods

Four similarity measure methods, namely, the single link, complete link, average link and Ward's method, were implemented. For comparison purpose, the same cut-off criterion was used for these four methods. The clusters were stopped to merge when the similarity measure between them is less than 0.5. The co-citation link strength threshold was set to 0.45. Table 5-8 shows the number of clusters obtained and the entropy values using these four clustering methods.

Table 5-8. Performance measurement of entropy using different clustering methods.

Clustering Method	***Number of Clusters Obtained***	***Entropy Value***
Single link	3	2.1673
Complete link	5	0.8562
Average link	3	2.5489
Ward's method	4	1.6932

The best performance is obtained using the complete link method. It produces tightly bound or compact clusters (Baeza-Yates, 1992). The single link method, in contrast, suffers from a chaining effect (Everitt, 1986) and produces the poorest

performance result. It has a tendency to generate clusters that are straggly or elongated. The average link method is very efficient when the objects form natural distinct "clumps" and it performs equally well with elongated, "chain" type clusters. The Ward's method is different from the above three methods because it uses an analysis of variance approach to evaluate the distances between clusters. In general, this method is quite efficient, however, it tends to create clusters in small size.

5.3 Summary

Author Co-citation Analysis is an effective method to generate author clusters according to the author co-citation data. But currently there is no system that makes use of co-citation to perform author clustering automatically. Researchers in the information studies field still rely one some statistical packages, such as SPSS, to get the statistics of the author co-citation patterns. Based on the output from these packages, researchers then manually group the authors into different research areas according to the authors' positions shown in the output. In our research, the author clusters are generated automatically using author co-citation analysis technique without the need of any human intervention.

CHAPTER 6
Conclusion and Future Work

6.1 Summary

With the enormous growth of the World Wide Web, different search engines or search tools have been developed to help people to locate their interested information in the Web. Currently, there are no search engines specifically for scientific publication retrieval as scientific literature normally appears in Portable Document Format (PDF) or PostScript formats, which are not indexed by most commercial search engines. This research tackles this problem by proposing and developing a scientific publication indexing and retrieval system known as PubSearch. It consists of three major components: Citation Indexing Agent, Web Citation Database and Intelligent Retrieval Agent. The Citation Indexing Agent searches through the Internet to locate the possible Web sites that contain scientific publications, and then browse through these Web sites to parse the scientific literature. The citation information is then extracted and stored in the Web Citation Database. The thesis focuses on discussing the Intelligent Retrieval Agent, which applies data mining techniques for document clustering and author clustering.

In the related work, some of the existing intelligent agents are firstly reviewed. To our knowledge, CiteSeer is the only system that supports automatic indexing and retrieval of scientific publications. Different from CiteSeer, we focus on mining the Web Citation Database for document clustering and author clustering for intelligent retrieval. For document clustering, both hierarchical and non-hierarchical clustering

techniques are investigated. In this research, the Kohonen's Self-Organizing Map (KSOM) and Fuzzy Adaptive Resonance Theory (Fuzzy ART) methods are selected as the mining techniques for document clustering. For author clustering, different techniques including Cluster Analysis, Multidimensional Scaling (MDS), and Factor Analysis are reviewed. The Agglomerative Hierarchical Clustering (AHC) of Cluster Analysis is used to generate author clusters. It is also combined with MDS to display author cluster maps.

The Web Citation Database consists of two major tables, SOURCE and CITATION. The SOURCE table stores the information of the source papers while the CITATION table stores all the citations extracted from the source papers. For the purpose of testing the mining results, a test citation database is used. This database is created by downloading the publications from 1987 to 1997 in the Information Retrieval field of the Social Science Citation Index from the Institute for Scientific Information's Web site, which includes all the journals on Library and Information Science. A total of 1,466 Information Retrieval related papers were selected from 367 journals with 44,836 citations.

For document clustering, the data mining process consists of five steps, namely, feature selection, pre-processing, transformation, document cluster generation, and retrieval. As the Web Citation Database only stores the citation information and no full-text is available, the traditional TFIDF approach cannot be adopted here. Generally, the more the two documents share the same citations, the more they are similar. Thus, document vectors can be formed by extracting keywords from the citations. KSOM and Fuzzy ART are implemented as two mining techniques to cluster documents. Performance results have shown that KSOM can achieve higher clustering and retrieval accuracy than Fuzzy ART. However, Fuzzy ART is more suitable for the

dynamic Web environment as re-leaning is not needed whenever there are new information added into the citation database.

For author clustering, the data mining process is based on co-citation analysis. It consists of six steps, namely, create author co-citation pairs, create raw co-citation matrix, convert into correlation matrix, generate author clusters, display author cluster map, and author retrieval. The AHC algorithm is used as the mining technique. The MDS technique is combined with the AHC algorithm to generate the author cluster maps. Four different methods, namely, the single link, complete link, average link, and Ward's method, are implemented and compared. The best performance is achieved using the complete link method. The author clusters from different time frame can also be used to track the change of the research area of a particular author or detect any newly emerged research fields.

In summary, this research (He and Hui, 2000) has achieved the following goals:

- A data mining process based on the KSOM and Fuzzy ART networks has been developed to mine the Web Citation Database for document clustering.
- A data mining process that incorporates the author co-citation analysis has been developed to mine the Web Citation Database for author clustering.
- An Intelligent Retrieval Agent has been designed and developed to support the retrieval of scientific publications over the WWW.

As a result, PubSearch has been developed. Document clustering as well as author clustering has been implemented. By supplying the appropriate keywords, users will get the full list of papers on the similar research topic even though some papers may not contain the exact keywords. In addition, users can also retrieve authors from the same research field by conducting the author search. There are "citing links" and

"cited links" options for every paper returned. By following these two options, users are able to retrieve all the papers cited by the current literature or all the papers that cite the current literature. Based on the number of citations the paper obtained, users will have an idea what is the impact of the paper in the scholarly community. The user interface of the PubSearch system is given in Appendix C.

6.2 Future Work

In this research, document clustering and author clustering have been investigated to mine the Web Citation Database. This research work can be further extended in various aspects, which are summarized below.

6.2.1 Combining Co-Citation Analysis with Co-Word Analysis

Co-word clustering focuses on the analysis of the titles or keywords used by authors (Zitt and Bassecoulard, 1994). The basic underlying assumption of this technique is that pairs of words that occur frequently together are statistically associated. This technique can be combined with the co-citation analysis technique in many ways.

One possible approach is that author co-citation analysis can be used to get author clusters. The publications by the authors within the same cluster are analyzed using the co-word analysis technique. By doing so, topics involved in a particular research area can be identified by aggregating a list of words together with their frequency of co-occurrence. Thus, a cluster word profile can be constructed to represent the research topic involved in the current work of the authors in the same cluster.

Another possible approach is that document co-citation can be used to get the document clusters. Then, co-word analysis technique is applied to the document clusters to generate the word profiles for each cluster. If document clusters identified by the co-citation analysis from different years have the similar word profiles, then they can be considered as different phases of the same research area. That is, combining co-citation and co-word analysis can be used to track the research areas over time.

6.2.2 Other Data Mining Tasks

Currently, the Web Citation Database is mined for document clustering and author clustering. Other mining tasks can also be performed on the citation database. Some possible mining tasks are listed as follows:

- *Identify the expert of one particular research area*. After generating author clusters, it is possible to calculate the frequency that one author's works cited by other scholars in the same cluster by looking into the citation information stored in the CITATION table.
- *Predict the trend of a specific research area*. The citation records can be grouped according to the "publication_date" of CITATION table falling into different periods. Then, cluster analysis can be performed on these groups separately. The historical data on the change of the research fields will be gathered, which can be used to predict the trends of various research areas.
- *Categorize journals or locate the leading journal*. Author co-citation analysis is used in author clustering. Journal co-citation analysis can also be conducted in a similar way. By doing so, journals can be categorized into different groups. Therefore, top journals can be identified.

6.2.3 Visualization of Results

The presentation of search results can be improved to be more intuitive using visualization and text mining techniques. For example, currently, the cluster map that returned by the KSOM neural network is just an array of cluster numbers. It is not clear what are these cluster numbers representing. To make the cluster maps easier to understand, text mining techniques can be used to analyze each cluster to extract several keywords that can describe the cluster concisely. Then, the description of each cluster can be shown in the cluster map and the relative distance between clusters may also be displayed.

6.2.4 Online Recommendation for New Updates

During the retrieval process, users' browsing behaviour can be recorded in a user profile. A centralised database can be established to store all the users' profiles. Clustering techniques can then be applied to this centralised database to classify users into different groups based on their browsing interests. Whenever there are any new publications added into the Web Citation Database, this publication will be analysed to locate the user group that has the highest possibility to have interest in it. Then, a recommendation can be made such that the new publication is broadcast to every user within the group.

References

ACM (2000). Association for Computing Machinery Digital Library Search. Available at <URL: http://www.acm.org/dl>.

Aggarwal C., Gates S. and Yu P. (1999). On the Merits of Building Categorization System by Supervised Clustering. In *Proceedings of the 5th ACM SIGKDD International Conference on Kowledge Discovery and Data Mining*, pp. 352 – 356,

Agrawal R. and Srikant R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 478-499.

Agrawal R., Faloutsos C. and Swami A. (1993). Efficient Similarity Search in Sequence Databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, pp. 69-84.

AltaVista Company (2000). AltaVista Search Home. Available at <URL: http://www.altavista.com >.

Arens Y., Knoblock C. and Shen W. (1996). Query Reformulation for Dynamic Information Integration. *Journal of Intelligent Information Systems*, Vol. 6, No. 2, pp. 99-130.

Armstrong R., Freitag D., Joachims T. and Mitchell T. (1995). WebWatcher: A Learning Apprentice for the World Wide Web. In *Proceedings of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*. AAAI Press.

At Home Co. (2000). Excite. Available at <URL: http://www.excite.com>.

Atzeni P., Mecca G. and Merialdo P. (1997). *Design and Maintenance of Data-Intensive Web Sites*. Technical Report 25, Department of Information Science, University of Rome. Available at <URL: http://www.dia.uniroma3.it/Araneus/articles.html).

Baeza-Yates R. A. (1992). Introduction to Data Structures and Algorithms Related to Information Retrieval. In *Information Retrieval: Data Structures and Algorithms*, pp. 13-27. New Jersey: Prentice-Hall, Inc.

Baldonado M. and Winograd T. (1997). SenseMaker: An Information-Exploration Interface Supporting the Contextual Evolution of a User's Interestes. In *Proceedings of Computer Human Interaction*, pp. 11-18.

Bellardo T. (1980). The Use of Co-Citations to Study Science. *Library Research*, Vol. 2, pp. 231-237.

Berry M. J. A. and Linoff G. (1997). *Data Mining Techniques*. New york: John Wiley & Sons, Inc.

Blair D. C. and Maron M. E. (1985). An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. *Communications of the ACM*, Vol. 28, pp. 291-230.

Boley D. (1998). Principal Direction Divisive Partitioning. *Data Mining and Knowledge Discovery*, Vol. 2, No. 4, pp. 325-344.

Bollacker K., Lawrence S. and Giles C. (1998). CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications. In *Proceedings of the 3rd ACM Conference on Digital Libraries*, pp. 116-123. Pittsburgh, PA.

Bollacker K., Lawrence S. and Giles C. (2000). Discovering Relevant Scientific Literature on the Web. *IEEE Intelligent Systems*, Vol. 15, No. 2, pp. 42-47.

Breiman L., Friedman J., Olshen R. and Stone C. (1984). *Classification of Regression Trees*. Wadsworth.

Carpenter G. and Grossberg S. (1987a). A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computer Vision, Graphics, and Image Processing*, Vol. 37, pp. 54-115.

Carpenter G. and Grossberg S. (1987b). ART2: Stable Self-Organization of Pattern Recognition Codes for Analog Input Patterns. *Applied Optics*, Vol. 26, pp. 4919-4930.

Carpenter G. and Grossberg S. (1990). ART3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures. *Neural Networks 3*, pp. 129-152.

Carpenter G., Grossberg S. and Rosen D. (1991). Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by An Adaptive Resonance System. *Neural Networks*, Vol. 4, pp. 759-771.

Chatfield C. and Collins A. J. (1989). *Introduction to Multivariate Analysis*. New York: Chapman and Hall.

Chawathe S., Garcia-Molina H., Hammer J., Ireland K., Papakonstantinou Y., Ullman J. and Widom J. (1994). The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *Proceedings of IPSJ Conference*, pp. 7-18. Tokyo, Japan.

Cheeseman P. and Stutz J. (1996). Bayesian Classification (AutoClass): Theory and Restuls. In Fayyad U. M., Piatetsky-Shapiro G., Smyth P. and Uthurusamy R., editors, *Advances in Knowledge Discovery and Data Mining*, pp. 153-180. AAAI/MIT Press.

Chen C. (1999). Visualising Semantic Spaces and Author Co-Citation Networks in Digital Libraries. *Information Processing & Management*, Vol. 35, No. 3, pp. 401-420.

Chen C. and Carr L. (1999). Trailblazing the Literature of Hypertext: Author Co-Citation Analysis (1989-1998). In *Proceedings of the 10th ACM Conference on Hypertext and Hypermedia: Returning to Our Diverse Roots (Hypertext '99)*, pp. 51-60. Darmstadt, Germany.

Croft W. (1980). A Model of Cluster Searching Based on Classification. *Information Systems*, Vol. 5, pp. 189-195.

Cronin B. and Snyder H. (1997). Comparative Citation Rankings of Authors in Monographic and Journal Literature: a Study of Sociology. *Journal of Documentation*, Vol. 53, No. 3, pp. 263-273.

Cutting D., Karger D., Pederson J. and Tukey J. (1992). Scatter/Gather: a Cluster-Based Approach to Browsing Large Document Collections. In *proceedings of ACM/SIGIR*, pp. 318-329.

Deerwester S., Dumais S., Furnas G. and Landauer K. (1990). Indexing by Latent Semantic Analysis. *Journal of American Society for Information Science*, Vol. 41, pp. 391-407.

Ding, Y. (1998). Visualization of Intellectual Structure in Information Retrieval: Author Co-Citation Analysis. *International Forum on Information and Documentation*. Vol. 23, No. 1, pp. 25-36.

Disney Enterprises Inc. (2000). Go.com. Available at <URL: http://www.go.com>.

Dörre J., Gerstl P. and Seiffert R. (1999). Text Mining: Finding Nuggets in Mountains of Textual Data. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 398 – 401.

Etzioni O. and Weld D. (1994). A Softbot-Based Interface to the Internet. *Communications of the ACM*, Vol. 37, No. 7, pp. 72-76.

Everitt, B. (1986). *Cluster Analysis* (2nd ed.). Hampshire: Gower Publishing Company Limited, England.

Faloutsos C. and Lin K. I. (1995). FastMap: A Fast Algorithm for Indexing, Data Mining and Visualization of Traditional and Multimedia Datasets. In *Proceedings of ACM SIGMOD*, pp. 163-174.

Fayyad U. (1998). Mining Database: Towards Algorithms for Knowledge Discovery. *Bulletin of the Technical Committee on Data Engineering*, Vol. 21, No. 1, pp. 39-48.

Fayyad U. M., Piatetsky-Shapiro G. and Smyth P. (1996). From Data Mining to Knowledge Discovery: An Overview. In Fayyad U. M., Piatetsky-Shapiro G., Smyth P., and Uthurusamy R., editors, *Advances in Knowledge Discovery and Data Mining*, pp. 1-34. AAAI/MIT Press.

Fisher D. (1987). Improving Inference through Conceptual Clustering. In *Proceedings of 1987 AAAI Conference*, pp. 461-465. Seattle, Washington.

Fisher D. (1995). Optimization and Simplification of Hierarchical Clusterings. In *Proceedings of 1st International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pp. 118-123. Montreal, Canada.

Forgy E. (1965). Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications. *Biometrics*, Vol. 21, pp. 768-769.

Fox E. A., Akscyn R. M., Furuta R. K. and Leggett J. J. (1995). Digital Libraries. *Communications of the ACM*, Vol. 38, No. 4, pp. 22-28.

Garfield E. (1980). ABCs of Cluster Mapping, Part 1, Most Active Fields in the Life Sciences in 1978. *Current Comments*, No. 40, pp. 5-12.

Garofalakis M., Rastogi R., Seshadri S. and Shim K. (1999). Data Mining and the Web: Past, Present and Future. In *Proceedings of the 2nd International Workshop on Web Information and Data Management*, pp. 43 – 47.

Giles C., Bollacker K. and Lawrence S. (1998). CiteSeer: an Automatic Citation Indexing System. In *Proceedings of the 3rd ACM Conference on Digital Libraries*, pp. 89-98. Pittsburgh, PA.

Gorsuch, R. (1983). *Factor Analysis* (2nd ed.). Hillsdale, NJ: Erlbaum.

Google Inc. (2001). *Google WebSearch*. Available at <URL: http://www.google.com/>.

Green P. E., Carmone F. J. and Smith S. M. (1989). *Multidimensional Scaling: Concepts and Applications*. Boston: Allyn and Bacon.

Grossberg S. (1986). The Adaptive Self-Organization of Serial Order in Behavior: Speech, Language and Motor Control, in Schwab E. and Nusbaum H., editors, *Pattern Recognition By Humans and Machines, Vol. I: Speech Perception*. Academic Press, Inc.

Gusfield D. (1997). *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*, Chapter 6. Cambridge University Press.

Hammond K., Burke R., Martin C. and Lytinen S. (1995). FAQ-Finder: A Case-Based Approach to Knowledge Navigation. In *Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogenous, Distributed Environments*. AAAI Press.

Han E., Boley D., Gini M., Gross R., Hastings K., Karypis G., Kumar V., Mobasher B. and Moore J. (1998). WebACE: A Web Agent for Document Categorization and Exploration. In *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 408–415.

Han J. (1997). OLAP Mining: An Integration of OLAP with Data Mining, In *Proceedings of 1997 IFIP Conference on Data Sematics (DS-7)*, pp. 1-11. Leysin, Switzerland.

Han J. (1999). Data Mining, in *Urban J. and Dasgupta P., editors, Encyclopedia of Distributed Computing*. Kluwer Academic Publishers.

Han J. and Fu Y. (1995). Discovery of Multiple-Level Association Rules from Large Databases. In *Proceedings of the 21st International Conference on Very Large Data Bases*, pp. 420-431.

Han J. and Fu Y. (1996). Exploration of the Power of Attribute-Oriented Induction in Data Mining. *Advances in Knowledge Discovery and Data Mining*, pp. 399-421. AAAI/MIT Press.

Han J., Cai Y. and Cercone N. (1993). Data-driven Discovery of Quantitative Rules in Relational Databases. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, pp. 29 – 40.

Harter S. P. (1992). Psychological Relevance and Information Science. *Journal of American Society for Information Science*, Vol. 43, pp. 602-615.

Hartigan J. A. 1975. *Clustering Algorithms*. New York: John Wiley and Sons, Inc.

He Y. and Hui S. C. (2000). Mining Citation Database for the Retrieval of Scientific Publications over the WWW. In *International Conference on Intelligent Information Processing*. Beijing, P. R. China.

Hertz J., Krogh A. and Palmer R. G. (1991). *Introduction to the Theory of Neural Computation*. Santa Fe Institute Studies in the Sciences of Complexity lecture notes. Addison-Wesley Longman Publication Corporation.

Herwijnen E. (1994). *Practical SGML (2nd ed.)*. Kluwer Academic Publishers.

Hill D. (1968). A Vector Clustering Technique. In Samuelson, editors, *Mechanized Information Storage, Retrieval and Dissemination*. North-Holland, Amsterdam.

Ho L. V. (2000). *Monitoring and Tracking Web Publications over the WWW*. M.A.Sc. Thesis, School of Computer Engineering, Nanyang Technological University, Singapore.

Honkela T., Kaski S., Lagus K. and Kohonen T. (1997). WEBSOM – Self-Organizing Maps of Document Collections. In *Proceedings of the Workshop on Self-Organizing Maps*, pp. 310-315. Espoo, Finland.

Honkela T., Kaski S., Kohonen T. and Lagus K. (1998). Self-organizing maps of very large document collections: Justification for the WEBSOM method. In Balderjahn I., Mathar R. and Schader M., editors, *Classification, Data Analysis, and Data Highways*, pp. 245-252. Springer, Berlin.

IEEE (2000). IEEE Digital Library Search. Available at <URL:http://www.computer.org/search.htm>.

ISI (2000). Institute for Scientific Information. Available at <URL: http://www.isinet.com>.

Jain A. K. and Dubes R. C. (1988). *Algorithms for Clustering Data*, Prentice-Hall advanced reference series. New Jersey: Prentice-Hall, Inc.

Jain A. K., Murty M. N. and Flynn P. J. (1999). Data Clustering: A Review. *ACM Computing Surveys*, Vol. 31, No. 3, pp. 264-323.

Jardine N. and Van Rijsbergen C. (1971). The Use of Hierarchic Clustering in Information Retrieval. *Information Storage and Retrieval*, Vol. 7, pp. 217-240.

Johnson, A. G. (1988). *Statistics*. Orlando, Florida: Harcourt Brace Jovanovich, Publishers.

Kaski S. (1998). Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering. In *Proceedings of International Joint Conference on Neural Networks (IJCNN'98*), Vol. 1, pp. 413-418. IEEE Service Centre, Piscataway, NJ.

Kaski S., Lagus K., Honkela T. and Kohonen T. (1998). Statistical Aspects of the WEBSOM System in Organizing Document Collections. *Computing Science and Statistics*, Vol. 29, pp. 281-290.

Kaufman L. and Rousseeuw P. (1990). *Finding Groups on Data: An Introduction to Cluster Analysis*. New York: John Wiley and Sons, Inc.

Kautz H., Selman B. and Shah M. (1997). Referral Web: Combining Social Networks and Collaborative Filtering. *Communications of ACM*, Vol. 40, No. 3, pp. 63-65.

Kerlinger F. (1973). *Foundations of Behavioral Research* (2nd ed.). New York; Holt, Rinehart & Winston.

King B. (1967). Step-Wise Clustering Procedures. *Journal of American Statistics Association*, Vol. 69, pp. 86-101.

Kirk T., Levy A., Sagiv Y. and Srivastava D. (1995). The Information Manifold. In *Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogenous, Distributed Environments*. AAAI Press.

Kohonen T. (1990). The Self-Organizing Map, In *Proceedings of IEEE*, Vol. 78, No. 9, pp. 1464-1480.

Kohonen T. (1995). *Self-Organizing Maps*. Springer.

Kohonen T. (1998). Self-Organizing of Very Large Document Collections: State of the Art. In *Proceedings of the 8th International Conference on Artificial Neural Networks*, Vol. 1, pp. 65-74. Springer, London.

Kohonen T., Kaski S., Lagus K., Salojärvi J., Paatero V. and Saarela A. (2000). Self-Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks*, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, Vol. 11, No. 3, pp. 574-585.

Konstan J. A., Miller B. N., Maltz D., Herlocker J. L., Gordon L. R. and Riedl J. (1997). GroupLens: Applying Filtering to Usenet News. *Communications of ACM*, Vol. 40, No. 3, pp. 77-87.

Krulwich B. and Burkey C. (1996). The ContactFinder Agent: Answering Bulletin Board Questions with Referrals. In *Proceedings of 13th National Conference of Artificial Intelligence (AAAI96)*, pp. 10-15. California: AAAI Press.

Kruskal J. (1977). The Relationship Between Multidimensional Scaling and Clustering. *Classification and Clustering*, pp. 17-44. New York: Academic Press.

LaMacchia B. (1996). *Internet Fish, a Revised Version of a Thesis Proposal*. MIT, AI Lab and Department of Electrical Engineering and Computer Science, Cambridge, Mass.

Lang K. (1995). News Weeder: Learning to Filter Netnews. In *Proceedings of 12th International Conference on Machine Learning*, pp. 331-339.

Lawrence S. and Giles L. (1999). Accessibility and Distribution of Information on the Web. *Nature*, Vol. 400, pp. 107-109.

Lawrence S., Giles C. and Bollacker K. (1999). Digital Libraries and Autonomous Citation Indexing. *IEEE Computer*, Vol. 32, No. 6, pp. 67-71.

Levy D. and Marshall C. (1995). Going Digital: a Look at Assumptions Underlying Digital Libraries. *Communications of ACM*, Vol. 38, No. 4, pp. 77-84.

Light R. (1997). Presenting XML. Indianapolis: Sams Net.

Lin, X. 1997. Map displays for information retrieval. *Journal of the American Society for Information Science*, Vol. 48, pp. 40-54.

Linde Y., Buzo A. and Gray R. (1980). An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, Vol. 28, pp. 84-95.

Lloyd S. (1982). Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, Vol. 28, pp. 129-137.

Lycos Inc. (2000a). Lycos. Available at <URL: http://www.lycos.com>.

Lycos Inc. (2000b). HotBot. Available at <URL: http://www.hotbot.com>.

Maarek Y. and Ben Shaul I. (1996). Automatically Organizing Bookmarks Per Content. In *Proceedings of 5th International World Wide Web Conference*, pp. 1321-1334.

McCain K. (1990). Mapping Authors in Intellectual Space: a Technical Overview. *Journal of the American Society for Information Science*, Vol. 41, No. 6, pp. 433-443.

Michalski R. (1983). A Theory and Methodology of Inductive Learning. *Machine Learning: An Artificial Intelligence Approach*, Vol. 1, pp. 83-134. Morgan Kaufmann.

Mitchell T. (1999). Machine Learning and Data Mining. *Communications of the ACM*, Vol. 42, No. 11, pp. 31-36.

Mitchell T., Caruana R., Freitag D., McDermott J. and Zabowski D. (1994). Experience with a Learning Personal Assistant. *Communications of the ACM*, Vol. 37, No. 7, pp. 80-91.

Mladenic D. and Institute J. (1999). Text-Learning and Related Intelligent Agents: A Survey. *IEEE Intelligent Systems*, Sep/Oct 99, pp. 44-54.

Moor B. K. (1988). ART 1 and Pattern Clustering. In *1988 Connectionist Summer School*, pp. 174-185. Morgan Kaufmann, San Mateo, CA.

Pao M. L. (1993). Term and Citation Retrieval: a Field Study. *Information Processing & Management*, Vol. 29, No. 1, pp. 95-112.

Park J., Chen M. and Yu P. (1995). An Effective Hash Based Algorithm for Mining Association Rules. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 175-186.

Pazzani M., Muramatsu J. and Billsus D. (1996). Syskill & Webert: Identifying Interesting Web Sites. In *Proceedings of 13th National Conference on Artificial Intelligence AAAI 96*, pp. 54-61. AAAI Press, Menlo Park, California.

Prescript (1998). *PreScript - A Utility for Extracting Text from PostScript Files*. Available at <URL: http://www.nzdl.org/html/prescript.html>.

Quinlan J. (1986). Induction of Decision Trees. *Machine Learning*, Vol. 1, pp. 81-106.

Quinlan J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Fransisco, USA.

Rao R., Pedersen J., Hearst M. A., Mackinlay J. D., Card S. K., Masinter L., Halvorsen P. and Robertson G. C. (1995). Rich Interaction in the Digital Library. *Communications of ACM*, Vol. 38, No. 4, pp. 29-39.

Rauber, A., and Merkl, D. 1999. SOMLib: A digital library system based on neural networks. In *Proceedings of the 4th ACM Conference on Digital Libraries (DL'99)*, pp. 240-241. Berkeley, CA.

Rocchio J. (1966). *Document Retrieval Systems – Optimization and Evaluation*. Ph.D. Thesis, Harvard University.

Rucker J. and Marcos J. (1997). Siteseer: Personalised Navigation for the Web, *Communications of ACM*, Vol. 40, No. 3, pp. 73-75.

Sahami M., Yusufali S. and Baldonado M. (1998). SONIA: A Service for Organizing Networked Information Autonomously. In *Proceedings of the 3rd ACM Conferences on Digital Libraries*, pp. 200-209.

Salton G. (1991). Developments in Automatic Text Retreival. *Science*, Vol. 253, pp. 974-979.

Salton G. and McGill M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill Publishing Company.

Saracevic T. (1995). Evaluation of Evaluation in Information Retrieval. In *Proceedings of the 18th ACM SIGIR Conference on Research and Development of Information Retrieval*, pp. 138-146. Seattle.

Saracevic T. (1996). Relevance Reconsidered. *Information Science: Integration in Perspective*, pp. 210-218. Copenhagen: Royal School of Librarianship.

Schatz B. and Chen H. (1996). Building Large-scale Digital Libraries. *IEEE Computer*, Vol. 29, No. 5, pp. 22-26.

Schvaneveldt R. W., Durso F. T. and Dearholt D. W. (1989). Network Structures in Proximity Data. In Bower G., editors, *The Psychology of Learning and Motivation*, Vol. 24, pp. 249-284. Academic Press.

Shavlik J. and Eliassi-Rad T. (1998). Building Intelligent Agents for Web-based Tasks: a Theory-refinement Approach. *Working Notes of Learning from Text and the Web, Conference Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh. Available at <URL: http://www.cs.cmu.edu/~conald/conald.html>.

Shin D., Jang H. and Jin H. (1998). BUS: An Effective Indexing and Retrieval Scheme in Structured Documents. In *Proceedings of the 3rd ACM Conferences on Digital Libraries*, pp. 235-243.

Slonim N. and Tishby N. (2000). Document Clustering using Word Clusters via the Information Bottleneck Method. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 208-215. Athens, Greece.

Small H. and Griffith B. (1974). The Structure of Scientific Literatures, I: Identifying and Graphing Specialties. *Science Studies*, Vol 4. No. 17, pp. 339-365.

Sneath P. H. A. and Sokal R. R. (1973). *Numerical Taxonomy*. London: Freeman.

SPSS Inc. (2000). *Statistical Package for the Social Sciences*. Available at <URL: http://www.spss.com).

Srikant R. and Agrawal R. (1995). Mining Generalized Association Rules. In *Proceedings of the 21st International Conference on Very Large Data Bases*, pp. 407-419.

The Dialog Co. (2001). The Dialog Corporation Home Page. Available at <URL: http://www.dialog.com>

Tishby N., Pereira F. C. and Bialek W. (1999). The Information Bottleneck Method. In *Proceedings of 37th Allerton Conference on Communication, Control and Computing*.

Turtle H. and Croft W. B. (1991). Evaluation of an Inference Network-Based Retrieval Model. *ACM Transactions on Information Systems*, Vol. 9, No. 3, pp. 187-222.

Van Rijsbergen C. (1974). Further Experiments With Hierarchic Clustering in Document Retrieval. *Information Storage and Retrieval*, Vol. 10, pp. 1-14.

Van Rijsbergen C. (1979). *Information Retrieval (*2nd ed.). Utterworths, London, England.

Voorhees E. (1986). Implementing Agglomerative Hierarchical Clustering Algorithms for Use in Document Retrieval. *Information Processing and Management*, Vol. 22, No. 4, pp. 65-76.

Ward J. H. Jr. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of American Statistics Association*, Vol. 58, pp. 236-244.

WestGroup (2000). Keycite – A Powerful Citer for Citation Search. Available at <URL: http://www.westgroup.com/products/keycite>.

White H. and Griffith B. (1981). Author Co-Citation: A Literature Measure of Intellectual Structure. *Journal of the American Society for Information Studies*, Vol. 32, pp. 163-171.

White H. and McCain K. (1998). Visualizing a Discipline: An Author Co-Citation Analysis in Information Science, 1992-1995. *Journal of the American Society for Information Science*, Vol. 49, pp. 327-356.

Will T. (1999). *Introduction to the Singular Value Decomposition*. Available at <URL: http://www.davidson.edu/math/will/svd/index.html>.

WordNet (2000). *WordNet – A Lexical Database for English*. Available at <URL: http://www.cogsci.princeton.edu/~wn).

Yahoo! Inc. (2000). Yahoo!, Available at <URL: http://www.yahoo.com>.

Yang Y. and Liu X. (1999). A Re-Examination of Text Categorization Methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 42-49.

Zamir O. and Etzioni O. (1998). Web Document Clustering: a Feasibility Demonstration. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 46 – 54.

Zitt M. and Bassecoulard E. (1994). Development of a method for detection and trend analysis of research fronts built by lexical or co-citation analysis. *Scientometrics*, Vol. 30, No. 1, pp. 333-51.

Appendix A 50 Queries for Performance Evaluation on Document Clustering

1. Relationship between recall and precision.
2. Data mining applied on information retrieval.
3. Intelligent systems for information retrieval.
4. Information retrieval using neural networks.
5. Artificial intelligence and information retrieval.
6. Latent semantic indexing and vector space model.
7. Performance measurement for information retrieval.
8. Trends in research on information retrieval.
9. Use of thesaurus structure in information retrieval.
10. Evaluation of the user interface in information retrieval system.
11. Evaluation issues in information retrieval.
12. Knowledge representation methods and algorithms.
13. Query formulation and expansion.
14. Retrieval of structured text information.
15. Information retrieval on scientific publications.
16. Telecommunication networking in online retrieval.
17. Artificial intelligence systems for chemical information retrieval.
18. Probabilistic models for information retrieval.
19. Generic algorithms and relevance feedback.
20. Natural language processing.
21. Data modeling and visualization.
22. Data storage technologies.
23. Database management and design.
24. Classification and clustering algorithms.
25. The relationship between indexing, structural linguistics and information retrieval.
26. User profiles generation and personalization of information retrieval system.
27. Graphical display of search results.

28. Digital image representation of retrieval.
29. Neural network models used for information systems.
30. Multimedia documents retrieval.
31. Client-server technology.
32. Retrieval of clinical science information.
33. Boolean query reformulation.
34. Document ranking methods.
35. Automatic text structuring and summarization.
36. Information retrieval in digital libraries.
37. Economics of information technology.
38. Automatic construction of hypertext.
39. Chinese text segmentation methods.
40. Citation searching and retrieval.
41. Document-focused and query-focused relevance feedback.
42. Human-computer interaction.
43. Distributed information retrieval systems.
44. Database management systems.
45. Cross-language information retrieval.
46. Information retrieval in psychology.
47. Database design for information retrieval.
48. Learning technique for hypertext categorization.
49. Semantic approach to extract classification knowledge of Internet documents.
50. Relevance feedback retrieval.

Appendix B Sample Data on Relevance Measurement for Document Clustering

The sample data on relevance measurement for one search session where the query is “intelligent systems for information retrieval” are given in this appendix. Table B-1 and Table B-2 list the system-based relevance and user-based relevance results using Kohonen’s Self-Organizing Map (KSOM) and Fuzzy Adaptive Resonance Theory (Fuzzy ART) algorithms respectively.

The “Rank Order” column shows the system ranking sequence. Only the first twenty documents retrieved by the system are considered. The “System-Based Relevance” column lists the relevance results calculated by the system. There are 10 sub-columns under the “User-Based Relevance” column, which store the relevance results evaluated by ten judges. The precision is calculated for both system-based and user-based relevance by averaging the individual relevance results accordingly.

Table B-1. System-based and user-based relevance results for KSOM.

Table B-2. System-based and user-based relevance results for Fuzzy ART.

Appendix C Retrieval User Interface of PubSearch

The scientific publications indexing and retrieval system, PubSearch, provides three different retrieval methods, namely, simple Boolean keyword search, document clustering search (based on Kohonen's Self-Organizing Maps (KSOM) and Fuzzy Adaptive Resonance Theory (Fuzzy ART)), and author clustering search (based on author co-citation analysis). For simple Boolean keyword search option, users are allowed to search paper title, author name, journal name, and publication date. The other two retrieval options have been discussed in Chapter 4 and Chapter 5 respectively.

C.1 Simple Keyword Search

The simple keyword search option will return the publications that contain the user-specified keyword(s). In addition to the search for keywords, users are also allowed to search documents based on author name, publication date, or journal name. Figure C-1 shows the user interface for searching the publication date and Figure C-2 displays the search results. All the titles of the listed papers are underlined to indicate the URL links provided. By clicking the paper title, the full-text of the selected paper will be displayed. Figure C-3 shows the full-text of the first paper from the result listing, which is entitled "On the Use of Information Retrieval Techniques for the Automatic Construction of Hypertext".

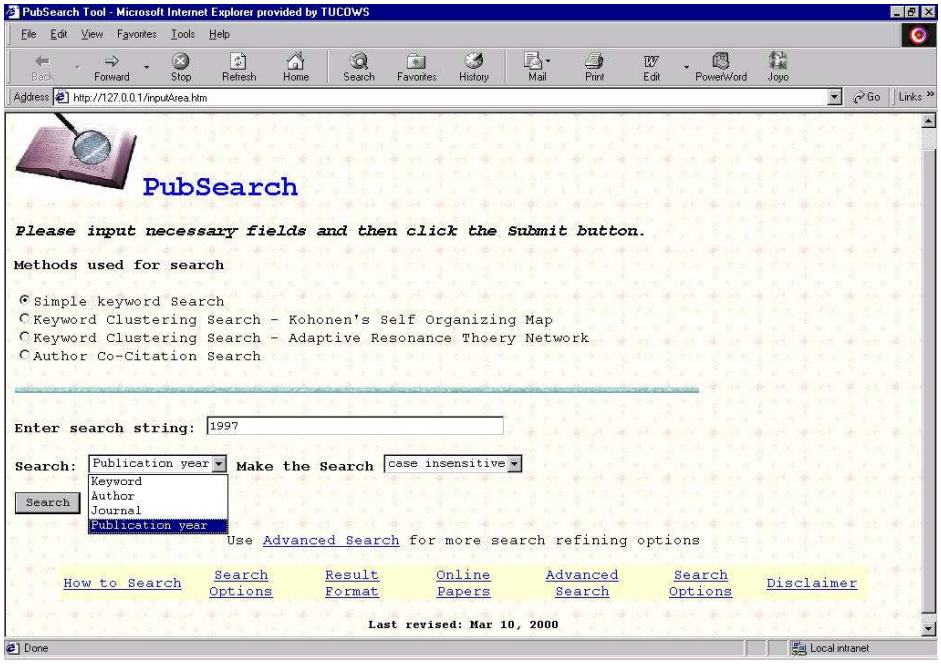


Figure C-1. Simple keyword search on “publication year = 1997”.

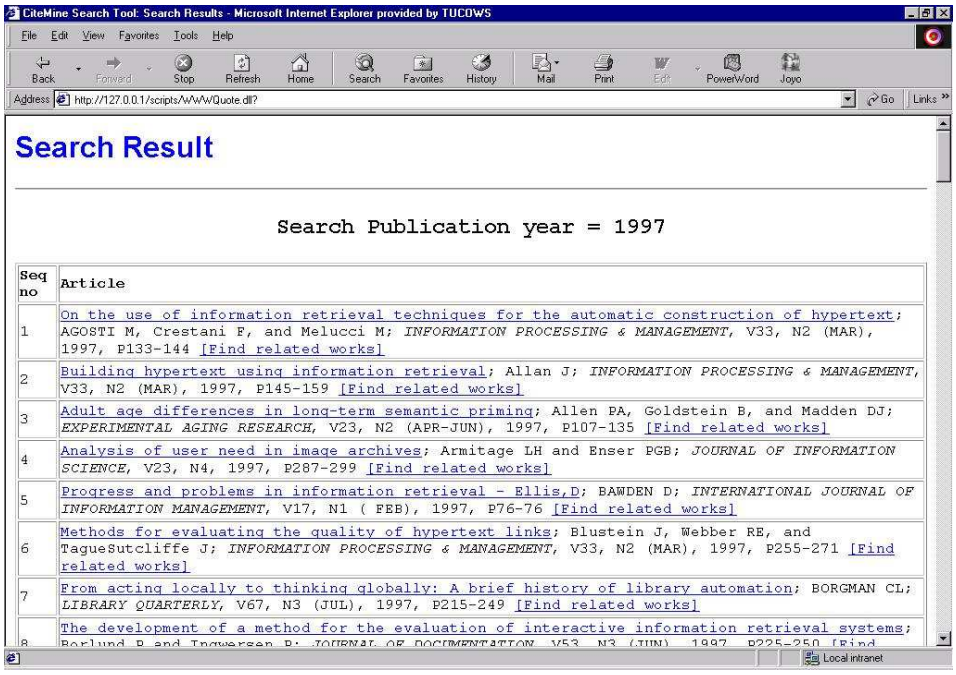


Figure C-2. Search result for simple keyword search on “publication year = 1997”.

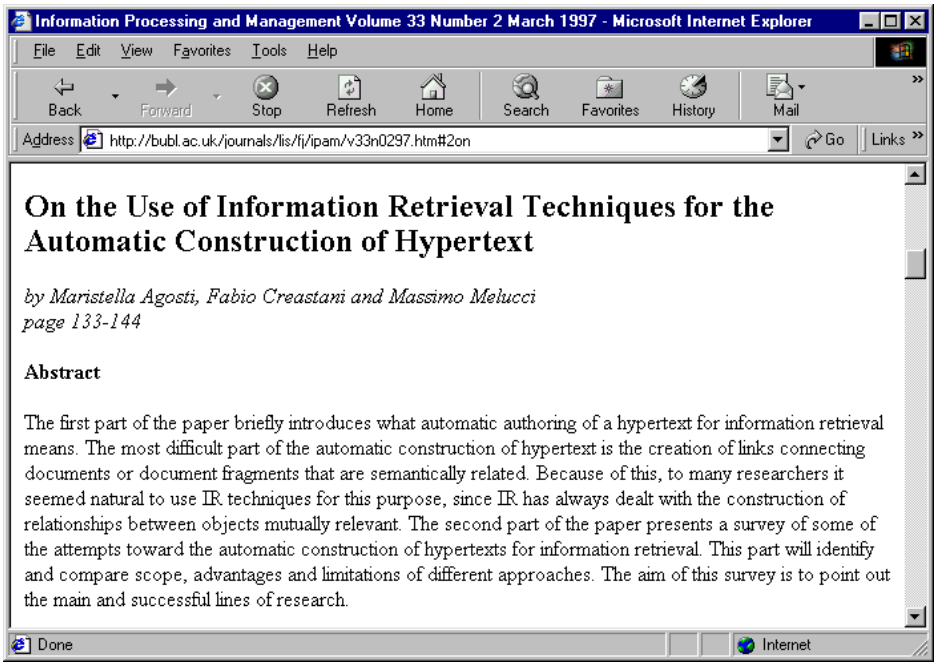


Figure C-3. Full-text of the paper – "On the Use of Information Retrieval Techniques for the Automatic Construction of Hypertext".

C.2 Document Clustering Search

PubSearch uses two different algorithms to categorize documents, KSOM and Fuzzy ART. Therefore, two options for document clustering search are provided.

Figure C-4 shows the user interface to search the query "knowledge based medical imaging". The search method chosen is "Document Clustering Search – Kohonen's Self-Organizing Map". PubSearch returns the cluster map as presented in Figure C-5. Cluster 85 is specified as the best match to the input query. By clicking the cluster number 85, the publications that are categorized under this cluster are listed as shown in Figure C-6.

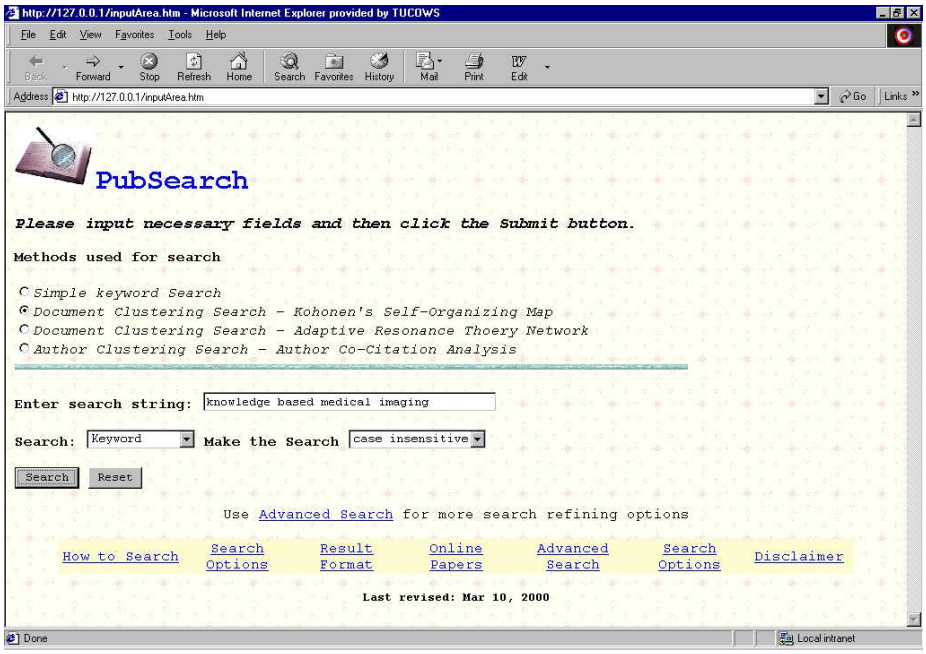


Figure C-4. Document clustering search based on KSOM.

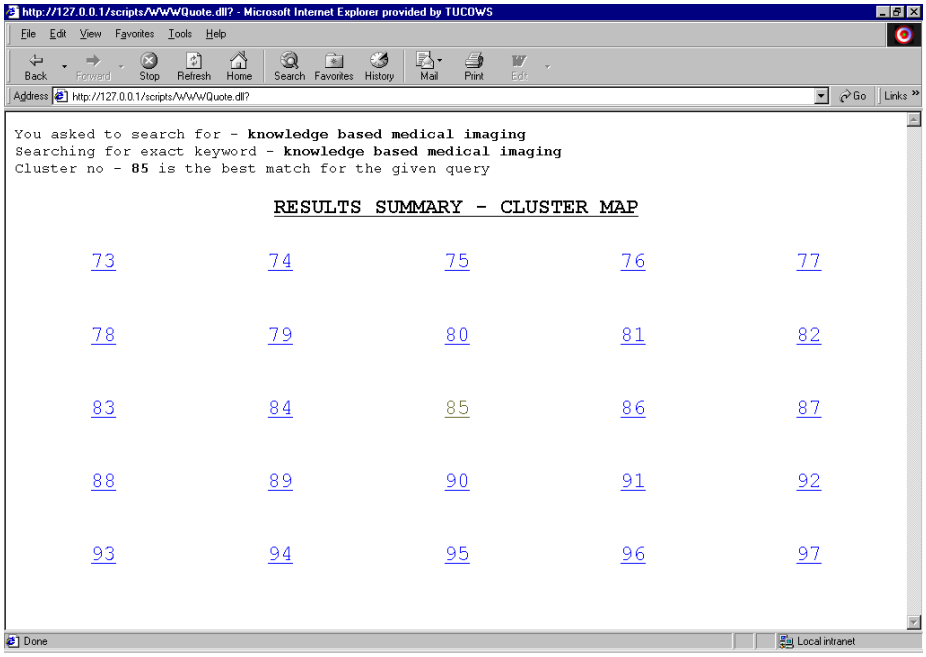


Figure C-5. Cluster map for the query “knowledge based medical imaging”.

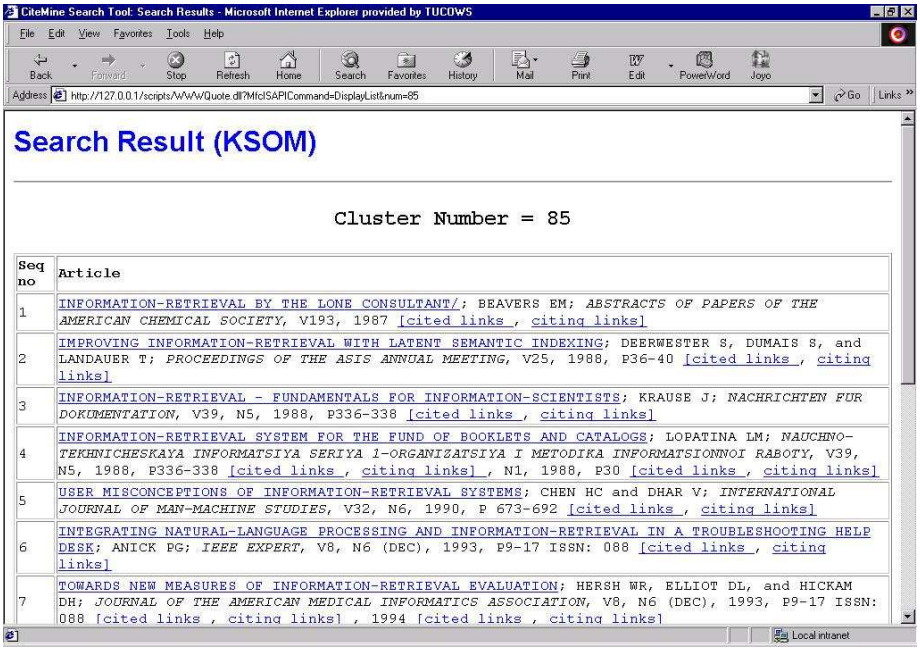


Figure C-6. Search result for the cluster number 85.

Figure C-7 shows the results obtained for the same query “knowledge based medical imaging” using the document clustering search based on Fuzzy ART option. It can be easily observed that the result listed here is different from the one obtained using KSOM.

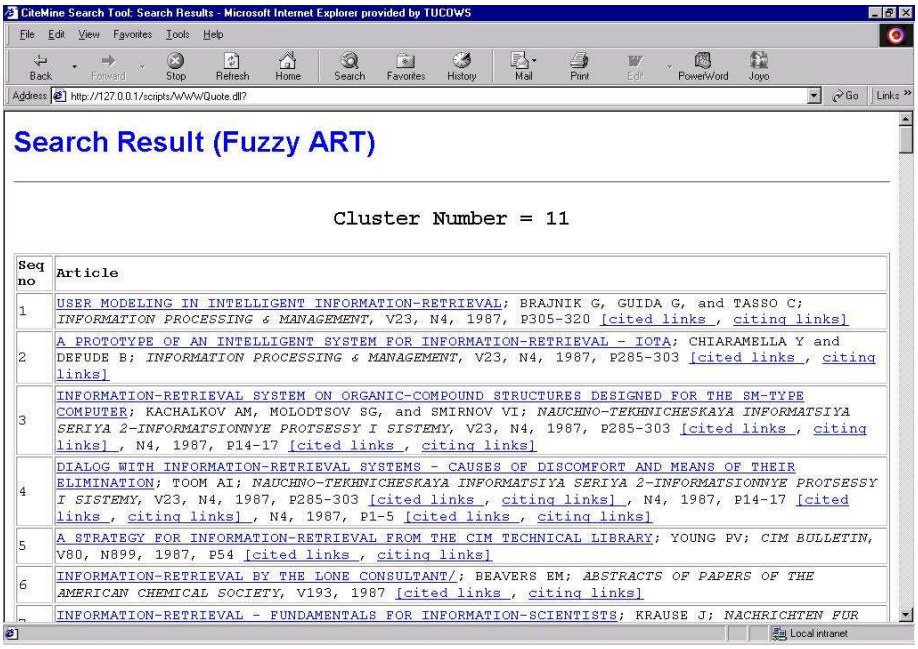


Figure C-7. Document clustering search using Fuzzy ART.

C.3 Author Clustering Search

Figure C-8 shows the user interface of PubSearch for author clustering search. The option “Author Co-Citation Search” has been chosen. The author name to be searched is “Belkin”. Figure C-9 shows the map of the cluster that the author “Belkin” belongs to. Each point represents an author. The distance between authors roughly corresponds to the similarity among them. By clicking any author names, the whole list of papers written by that author is displayed as shown in Figure C-10. All paper titles are underlined. By clicking any of the paper titles, the full-text of the publication will be displayed.

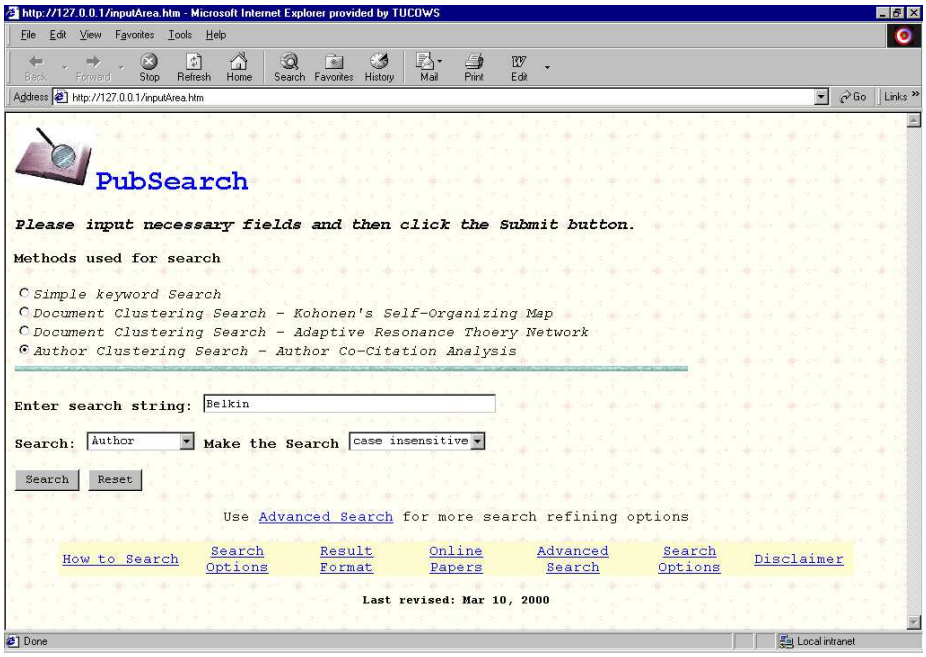


Figure C-8. PubSearch – Author Clustering Search.

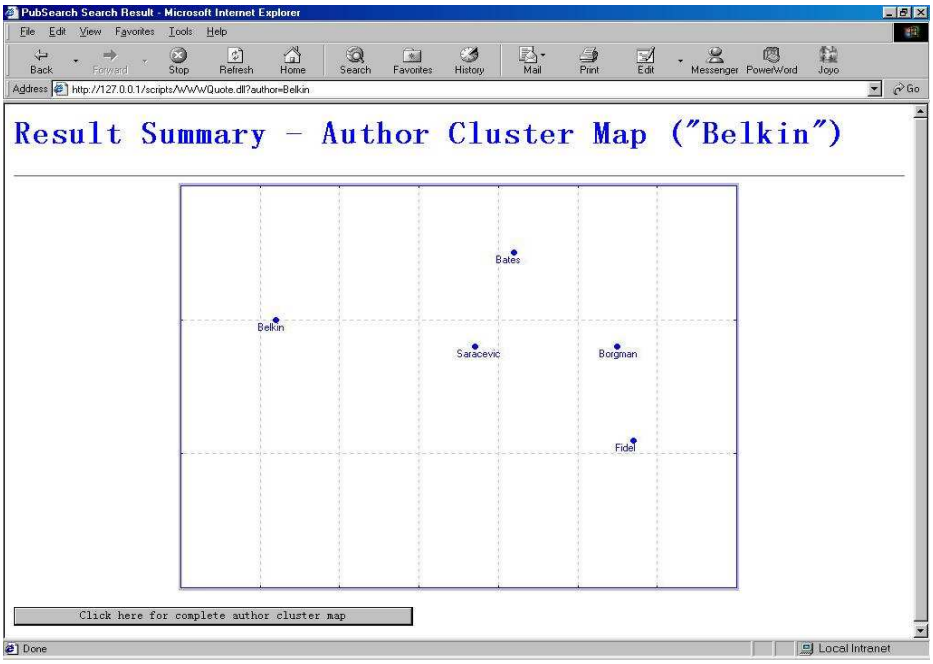


Figure C-9. Author cluster map for the query on Author = "Belkin".

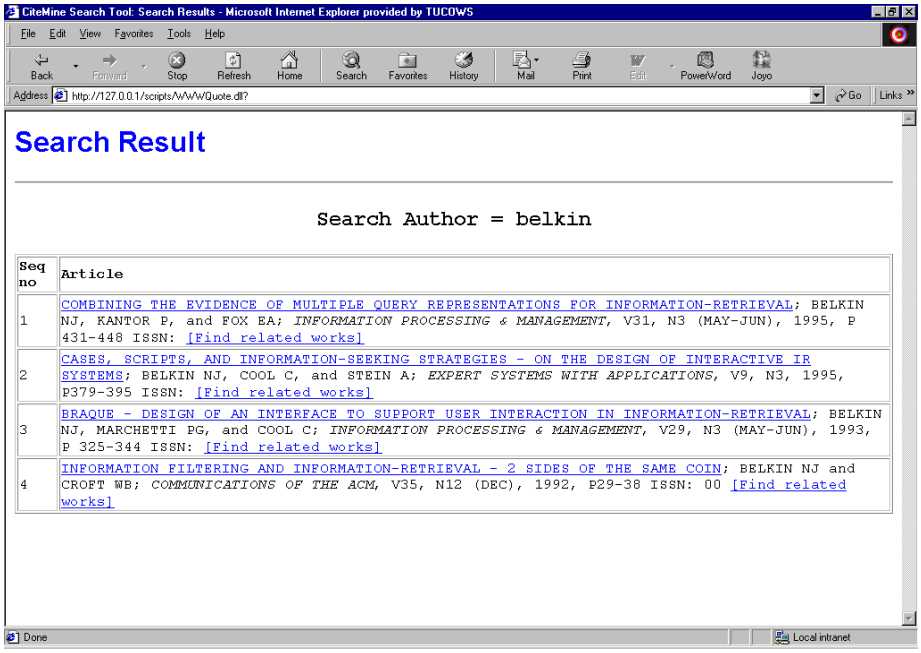


Figure C-10. Search result of "Author = Belkin" by Author Clustering Search.

Users are also allowed to view the overall author cluster map by clicking the button "Click here for overall author cluster map" as shown in Figure C-11. The author cluster map is illustrated in Figure C-11. Authors from different clusters are presented using the points with different colors or shapes.

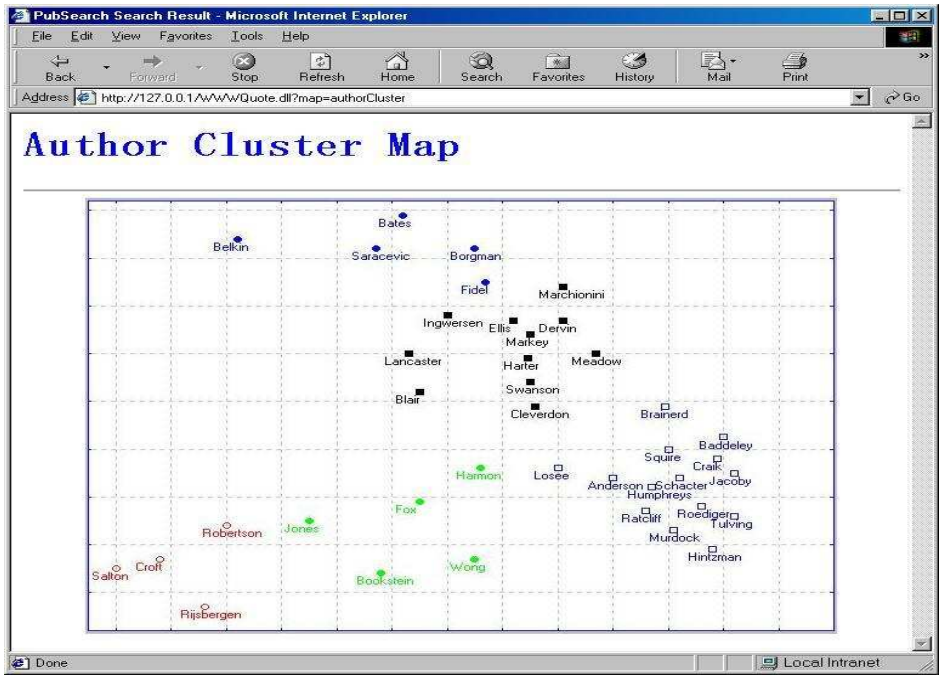


Figure C-11. Author cluster map.