AUTOMATIC SCIENTIFIC LITERATURE CLASSIFICATION USING MULTIPLE INFORMATION SOURCES FOR DATA MINING PURPOSES

By

BENYAM ASNAKE
Bachelor of Science
Addis Ababa University
Addis Ababa, Ethiopia
2000

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for the Degree of
MASTER OF SCIENCE
May, 2003

AUTOMATIC SCIENTIFIC LITERATURE CLASSIFICATION USING MULTIPLE INFORMATION SOURCES FOR DATA MINING PURPOSES

Thesis Approved:

____
Thesis Adviser

____

____

Timothy J. Pettibone
____
Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere appreciation to my major advisor, Dr. Gary Yen for his guidance, professional mentorship and friendly supervision. I would also like to thank Dr. Keith Teague and Dr. Rafael Fierro for serving in my graduate committee. Special thanks goes to Mr. Steven Morris for the useful comments, support and inspiring ideas. I am also very grateful for all the Intelligent Systems and Control Laboratory (ISCL) members, especially to Monica Wu, Michel Goldstein and Pedro de Lima for their constructive ideas and friendly support. Last but not least, I would like to thank my parents; my father Tesfaye Asnake and my mother Yewelsew Abebe for their love, guidance and Godly council.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

In the cutting age of technology the tremendous growth of information in the world has become one of the major concerns for those companies and institutions experiencing a continually increasing amount of data. Representative examples of these are the currently available digital libraries, research repositories, financial transactions, and governmental database systems that store various types of digital data including text, audio and image. This critical concern of information growth is not only an issue of data storage or management but also an issue of efficient utilization of stored data. One good example for this is the currently available web-search engines that return a list of information sources upon request to search for a specific keyword or query phrase. This result is generated based on a word-hit basis and the user has to go through every link to refine and extract the most relevant information that he or she is looking for. The obtained result is also highly dependent on prior knowledge about the subject matter and query phrase being used. But most of us might have wondered if there is really a means to automatically organize and classify those documents in an orderly manner so as to facilitate the searching process. Another good example is organization of a collection of journal articles according to their content, author, and the different collaboration groups within the collection so as to make their organization and presentation easy to understand and extract the most and hidden information out of them.

1.2 Motivation and Problem Statement

The motivation of this research stems from the idea of classifying a collection of journal documents without prior knowledge about their content and presenting them in a way that would reveal as much information as possible including the main research foci in the field of study, relationship between the documents, dominant authors, collaboration groups, and new emerging technologies.

The classification technique that is presented in this work is intended to help overcome or at least alleviate the traditional laborious part of text document organization and classification that almost every research involves. The methodology proposed in this work could also be extended to work for other types and format of data such as web pages and newspaper articles so as to help in solving the difficult link discovery and trend analysis problems that are currently demanding much of today's human intervention and expert knowledge.

1.3 Proposed Classification Architecture

Figure 1.1 below describes the classification architecture proposed in this work. It takes structured collection of journal articles as an input. Structured collection refers to a well-organized text data from a database in which all the specific fields such as the title, authors, references, key words are stored separately in an appropriate manner. This data is used to construct different similarity matrices that measure and represent the similarity between the documents. Each similarity matrix emphasizes on or coveys a particular type of relationship regarding the collection. For example, a similarity measure constructed from citation information leads to a good understanding on the trend of how people made use of previous knowledge. In a similar way, similarity information constructed from author data helps understand the different author collaboration groups within the research community.

In this work these different similarity measures are fused together using a set of fusion parameters to come up with a generalized similarity matrix that contains complement information extracted from different selected features. This matrix is later used to classify the documents in a more accurate way so as to extract the most hidden information out of the collection.

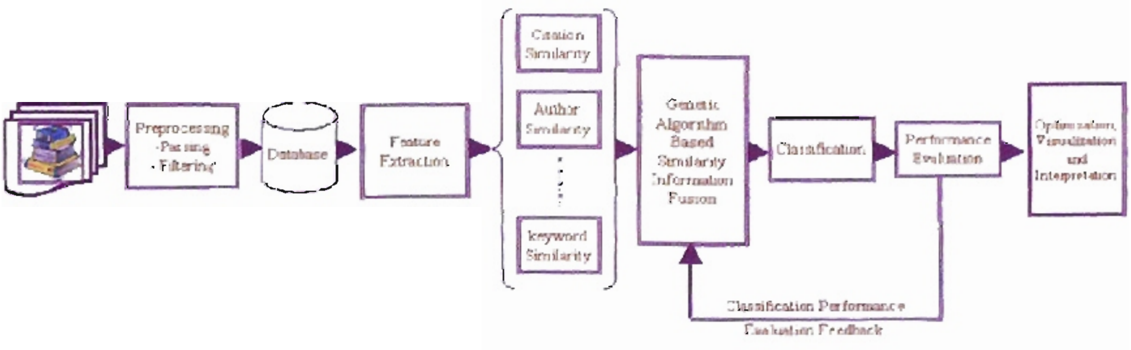


Figure 1.1 Text classification architecture used

The generalized similarity matrix is passed to an agglomerative hierarchical classification algorithm [1] in order to produce a user specified number of clusters. The performance of this classification is evaluated and fed back to the information fusion routine in order to search for the optimal set of the fusion parameters until a stopping condition is reached. When the stopping criterion is met, the best fusion parameters attained in the previous process are used to derive a generalized similarity matrix to cluster the data and the final clusters formed are visualized and interpreted using the two-dimensional time-line method presented in [2].

1.4 Methodology

The methodology followed in this research is as follows. First a set of articles of interest was collected from the ISI Science Citation Index library and saved as a text file. Then preprocessing was done on the collection and it was transformed into a Microsoft Access database using Visual Basic for Access routines. Information about each article was later extracted from this database to construct various types of similarity measures between the documents using a MATLAB program. A genetic algorithm was then employed to search for the best fusion parameters in order to combine these similarity matrices into a final generalized similarity matrix. This similarity matrix was used to perform hierarchical clustering. The result was visualized as a hierarchical time line that was optimized for visualization by using a simulated annealing based routine. The final result was used to explore and interpret the collection of the articles.

1.5 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2 the commonly used and dominant document classification and visualization methods will be discussed. Chapter 3 will cover how the similarity between documents can be measured and the different sources of similarity as applied to journal articles. Chapter 4 presents the proposed technique of fusing different similarity matrices in order to use them to classify the documents and the supporting visualization and interpretation technique used in this research. Chapter 5 discusses a case study using the techniques developed in this research. Final conclusions and discussion on future work is given in Chapter 6. Description of the programs and user interfaces used in this research are given in Appendix A.

CHAPTER II

LITERATURE REVIEW

2.1 Overview of Text Classification

The interest towards automatic scientific literature classification has been growing ever since the advent of fast computers and the enormous amount of text data growth being experienced in the world. The latter one being the major concern, today different text classification and categorization methods and solutions are developed for specific purposes aiming at minimizing and to the extent of avoiding the human effort in cataloging and classifying different types of text documents including web pages in the Internet, scientific literatures and books in digital libraries, news feeds, financial and governmental record databases, and more. These solutions intend to facilitate navigation, exploration, organization and presentation [3] of the documents despite the continuously growing size of the document collection.

One mainstream application area and that this research will also be devoted is the organization of text documents such as journal articles into groups on the basis of content similarity. Another example is automatic indexing of web pages like the YAHOO collection avoiding or even minimizing the use of human labor in creating those groups. Along this comes also the issue of fast and efficient computational requirements that will be mandatory as the size of the data grows.

So far different solutions across such types of problems involve utilization of similarity information from different sources such as available common links, citations, and word frequency similarity and application of appropriate supervised or unsupervised classification to this data.

2.2 Document Representation Models

Different models have been developed so far to represent documents and formulate query so as to facilitate information retrieval from a large document collection. These models help represent documents mathematically and can be used in information retrieval systems to produce results for a query by producing a ranked list of matches. They could also be employed to produce a generalized summary of similarity measure between each entity, which is a document in this case. For the completeness of the presentation, some of the most dominant information retrieval and document representation models are reviewed below.

2.2.1 The Vector Space Model

The vector space model [4] uses a selected feature, such as *terms* in documents, to represent documents. Documents and queries are then modeled as vectors in a term vector space if terms are selected as a representing feature. The frequency of each term would thus be a measure for a particular dimension resulting in a multi-dimensional way of representation of the documents. Figure 2.1 below illustrates this idea using a two-term space [4].

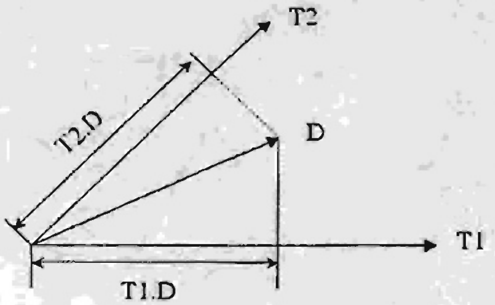


Figure 2.1 Document representation in a two-term space

In a more general way, a collection of large number of documents can be represented in terms of a matrix as shown below.

$$M = \begin{array}{c} \\ D_1 \\ D_2 \\ \cdots \\ D_n \end{array} \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & \cdots & T_t \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1t} \\ a_{21} & a_{22} & \cdots & a_{21} \\ \cdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nt} \end{bmatrix} \end{array} . \qquad (2.1)$$

Matrix M is an $n \times t$ matrix in which each row corresponds to a single document and each column to a particular term. $M_{i,j}$ represents the frequency of term j in document i. In other words, a document d would be represented as a vector as:

$$d_{tf} = (tf_1, tf_2, \ldots, tf_m), \qquad (2.2)$$

where tf_i represents the frequency of the i^{th} selected term. This kind of representation normally uses a term weighing scheme, as each term used in representing a document does not have equal significance. The most widely used weighing scheme in this kind of representation is what is known as *inverse-document frequency* (*IDF*) weighing scheme which gives more weight to those terms that have less frequency. This is mathematically represented as [3]:

$$d_{tf\_idf} = (tf_1 \times \log(\frac{N}{df_1}), tf_2 \times \log(\frac{N}{df_2}), \ldots, tf_m \times \log(\frac{N}{df_m})), \tag{2.3}$$

where df_i is the number of documents that contain term i and N is the total number of documents.

2.2.2 The Probabilistic Model

The vector space model assumes that terms are completely independent. The probabilistic model takes the inter relationship between terms into account. It also involves the inclusion of document relevance probability, i.e. each document is treated with a decreasing probabilistic relevance. Several versions of the Probabilistic Model have been developed that attempt to facilitate querying and information retrieval in large document collections [5]. Some of the variations among the different probabilistic models are displayed in the way terms and documents are treated. One variation of this model is displayed in the way the inter relationship between terms is considered. Some models consider only pair-wise dependency between terms while others extend this concept up to third or higher order of dependency [6]. Another variation is displayed in the way documents and queries are ranked by the model relevance measurement purposes [7].

2.2.3 The Boolean Model

This is the oldest and simplest model based on set theory that uses Boolean operators such as AND, OR, and NOT. It is most widely used in Internet search engines because of its fast computation. Query formulation plays an important role in this model since it retrieves exact match documents only. Several variations have been developed on this model to be able to rank documents. Among the major modifications are term weighting, usage of fuzzy operators [8] rather than just Boolean operators, and weighted query expansion using thesaurus [9].

2.3 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a technique introduced to improve the performance of information retrieval systems by overcoming the problem of synonymy and polysemy. Synonymy refers to different words having the same meaning and polysemy refers to same word having multiple meanings. In contrary to the assumption made in traditional information retrieval techniques that terms are independent, LSI models term-term inter relationship by mapping conceptually related terms closely in a semantic concept space [10]. This concept-space is a reduced version of the original *term by document* matrix and the dimensionality reduction is performed using the singular value decomposition [11]. LSI has also been applied to successfully retrieve information in a cross-lingual environment [12]. This is made possible by first training the LSI model with initial translated training documents in two languages and later adding in more documents in either language. Eventually a query phrase would be able to retrieve relevant documents in both languages [10].

2.4 Similarity Measures

The similarity information obtained from the above types of document representation models is a numeric representation of the measure of similarity between the entities. A similarity value of 1 would represent maximum or perfect similarity and a similarity value of 0 represents no similarity at all. From here onwards $S(i,j)$ will be used to represent the similarity between document i and document j and matrix S will be an $N \times N$ symmetric similarity matrix whose diagonal elements are all ones and the rest between zero and one.

The similarity between documents in such type of representation can be measured in two different ways [3]. One way is to consider each document as a unit vector and regard the cosine of the angle between the vectors as a similarity measure so that document vectors pointing in the same direction would have a similarity value of one and those orthogonal to each other would have a similarity value of zero. Mathematically, this can be represented as:

$$S(i,j) = \cos(d_i, d_j) = \frac{d_i^T d_j}{\|d_i\| \|d_j\|} = d_i^T d_j \,. \tag{2.6}$$

Another similarity measure is taking the inverse of the distance between the document vectors in such a way that documents close to each other would have less distance between them which would result in high similarity value and vise versa. Mathematically this can be represented as:

$$S(i,j) = \frac{1}{\text{distance}(d_i, d_j)} = \frac{1}{\|d_i - d_j\|}. \tag{2.7}$$

Among the different types of distance functions [13], the most widely used distance function in evaluating similarity between documents is the Euclidean distance defined as:

$$\|x - y\| = \sqrt{\sum_i (x_i - y_i)^2}. \tag{2.8}$$

The above type of *term-document* representation of n documents normally results in a high-dimensional matrix of size $n \times m$ where m tends to be very big. This information matrix also displays the nature of a sparse matrix, which would require additional care in terms of efficient memory utilization.

2.5 Dimensionality Reduction

The high dimensionality nature of the above types of document representation models results in an increased computational requirement and memory usage. The usual solution across this type 'curse of dimensionality' problem is normally dealt with by applying dimensionality reduction techniques such as feature selection, feature extraction and term or feature grouping [14].

2.5.1 Feature Selection

Feature selection reduces dimension by taking only the most discriminant features that would represent the dataset accurately to some extent with out major loss of information. For example, when representing documents in terms of the words that they contain, it is most of the time customary to have a list of stop words, such as *is, are, and*, etc., that do not actually have any particular meaning and also that are very frequent in almost all documents. This would help reduce the number of dimensions used in the final representation and save computational power. It also helps concentrate only on the unique and important features ignoring the most common ones that would further help identify each document uniquely from others.

2.5.2 Feature Extraction

Feature extraction uses different mapping techniques to represent the high dimensional data in terms of much lesser number of dimension vectors with out significant loss of information. Among the most common feature extraction techniques are principal component analysis (PCA) and random projection [15]. These are widely used mathematical methods that can be employed to reduce number of dimensions of a high-dimensional data into a fewer one.

2.5.3 Term Grouping

Term grouping is also another type of dimensionality reduction technique that helps reduce the number of features by combining the most similar features together. Each word in general has a particular root word that could give rise to many different ones. The technique of term grouping maps each word to its corresponding root word so as to reduce the total number of words used to represent a document. One way this could be achieved is by applying stemming to the prefix and suffix of words [16] and adding up their frequencies together. Another way of applying the concept of term grouping is through using a thesaurus in order to map terms that have similar meanings into one with out affecting the representation of the document.

CHAPTER III

DOCUMENT CLASSIFICATION

3.1 Document Similarity

The measure of similarity between documents is the key for the classification techniques that will be discussed afterwards. In this chapter different types of measures of similarity between documents are exploited. As mentioned in the previous chapter, the similarity between documents is a numeric value that would represent the measure of their similarity based on a selected feature and can be represented as a matrix. Each similarity matrix obtained from a selected feature conveys some aspects of the ideal similarity matrix that we will be trying to approximate. This concept is discussed in more detail in the next chapter, especially how different similarity matrices can be synergistically integrated to give the best approximation of the ideal similarity matrix.

In an *n*-dimensional feature space, say terms selected as a representing feature, the Euclidian distances between documents are inversely proportional to their corresponding similarity values. In other words, if documents are mapped to points in a high-dimensional feature space, those documents that appear close to each other would be more similar than those far from each other. Thus similarity is inversely proportional to distance, or distance can be considered as a measure of dissimilarity [17]. Transformation between these two can be done using different methods. However the most commonly used ones are shown below:

$$S(i,j) = \frac{1}{D(i,j)}, \tag{3.1}$$

or

$$S(i,j) = K - D(i,j), \tag{3.2}$$

where $S(i,j)$ and $D(i,j)$ are the similarity and dissimilarity/distance between documents i and j respectively. K is a constant selected based on the particular type of application.

When dealing with text documents the measure of similarity between documents largely depends on the type and format of the text data. For example, if the data is just a collection of plain text extracted from email messages, the only straightforward way of establishing similarity between the documents is using similarity information extracted from the terms in the documents. On the other hand if the collection of documents is a set of structured journal articles with all authors, institutions, citations, and keywords information available, then these features can give us different possibility of establishing similarity measure between the documents. This thesis will be concentrating only on journal articles in which all the above types of information are available in a well-structured manner.

The main challenge when using multiple sources of similarity information is on how to use them together in the final classification technique so as to produce a more accurate result. The method proposed to fuse these different similarity matrices and its results are discussed in detail in the next chapter. Below is given a detailed explanation on how to establish similarity between journal articles using different information sources that will be used later.

3.1.1 Citation Analysis

A well-documented citation data is a powerful source of information especially on tracing the trend of knowledge and information flow in a large collection of literature. Several works have been developed in utilizing citation information in areas of information retrieval and knowledge discovery, automatic library indexing [18], technology forecasting [19], and more. Citation information has proved to be helpful in different aspects of information retrieval and organization. A properly organized, citation-indexed system can provide useful information such as inter-document relationships, major improvements and criticisms of pervious work [18] and more. It can also provide useful information in identifying new emerging technologies that would be hard and time consuming to identify without such a system.

One good example for this is the Database Information and Visualization System (DIVA), a software tool developed in our research group, that was used to explore and visualize the US Patents database [19]. DIVA makes use of citation information of a selected set of *key patents* of user interest to build a larger collection of patents that are related to these key patents. A similarity matrix is then computed for this large collection based on citation information obtained from the database, which later is used to explore the inter-relationship between patents, including identifying the major contributions and forecasting future areas of development.

Citation information can be sub divided into four categories namely direct citation (dc), co-citation (cc), longitudinal coupling (lc), and bibliographic coupling (bc) of which the last three are types of indirect citation [20]. These types of citation are shown diagrammatically in Figure 3.1 below.

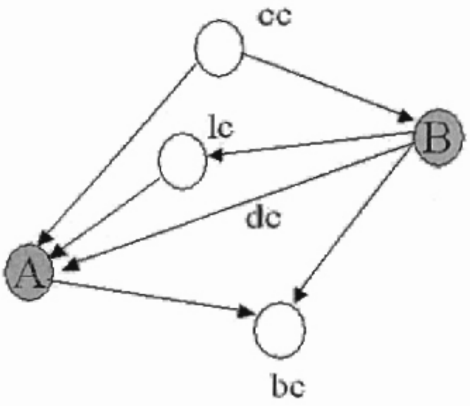


Figure 3.1 Example of different types of citations

Here the circles represent documents and the directed lines correspond to citation linkage. Documents *A* and *B* being the main focus of interest, the figure shows all the four types of direct and indirect citations. Depending on the type of application and information being provided, different types of combinational linkage could be formed by weighting and combining these different types of citation to represent the coupling or similarity between documents. However the method presented in this paper will consider only direct citation information only.

3.1.2 Author Co-Citation Analysis

Author's identity and affiliation conveys special information that helps in understanding a collection of documents since most researchers work within particular area(s) of research and collaboration groups. Author co-citation analysis (ACA) [21] has been introduced and studied for the past 15 years [22]. In this work the interest in ACA lies to exploit more information out of the collection that would help to better classify and further understand a collection of journal articles. ACA helps understand the relationship between different authors [23] and identify the different research collaboration groups in which these authors are associated with.

3.1.3 Word Frequency Analysis

The other useful feature that could be incorporated and used to classify a collection of text documents is term frequency. Though this sounds a straightforward idea at start, term frequency analysis demands a very thorough and detailed processing that most of the time requires extensive human interactions because the natural language is hard to automatically transform into perfect quantitative representation for a computer to process. Even so, several works have been carried out to categorize a collection of text documents into groups based on their word content analysis [14, 24, 25].

3.2 Construction of Document Similarity Matrix

Having discussed the above different sources of document similarity, this section will describe the mathematical details on how to construct similarity matrix out of such information. The first and foremost step towards constructing a similarity matrix from the above types of information is to form an adjacency matrix. An adjacency matrix, A, is a matrix that signals the presence or absence of a particular feature, such as term, author, or citation, in a document. It results into an $n \times m$ matrix that summarizes the relationship between n documents and m set of selected features. $A(i, j) = b$ means feature j appears b times in document i. In the case of citations and authors, A is a pure binary matrix, because an author can appear only once in a paper and a reference could be cited only once. Normally A tends to be a sparse matrix in which m is much larger than n and the need for dimensionality reduction arises before proceeding in order to speedup further computation.

After preprocessing is done on the original adjacency matrix the actual $n \times n$ similarity matrix that represents the inter document similarity can be computed using one of the following methods [4].

3.2.1 Inner Product

Representing documents and queries as vectors, an inner product between two vectors gives a value that represents how much the two vectors are similar to each other. In a two-dimensional vector space this can be illustrated as in Figure 3.2 below.

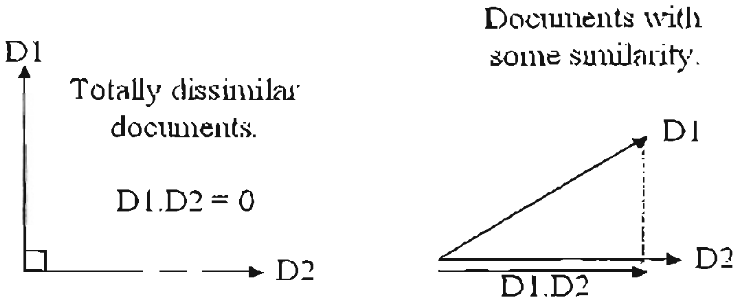

D1
Totally dissimilar documents.
D1.D2 = 0
D2
Documents with some similarity.
D1
D2
D1.D2

Figure 3.2 Inner-product similarity measure

From vector algebra, we know that the inner product of two vectors is zero if and only if they are orthogonal to each other. However this is less likely to happen in a large document space as documents usually share at least one common word. In the case of binary representation, this measures the number of co-occurring features in both documents. Mathematically the inner product between document vectors $D1$ and $D2$ is represented as:

$$Sim(D1,D2) = D1 \cdot D2 = \sum_{i=1}^{n} D1_i \cdot D2_i \qquad (3.3)$$

where n is the number of features to represent the documents and D_i correspond to the measure of the i^{th} feature.

Though this measure of similarity is easy in terms of computation there lies the hidden assumption in it that the features selected to represent documents are perfectly orthogonal or independent. However, this is not the case in practical applications. For example, most words in nature are related to each other and fail to satisfy this assumption.

3.2.2 Dice Coefficient

This similarity measure results in a number that lies within the range of [0,1]. This is a desirable feature in document representation because similarity values are usually normalized and it is compliant with this respect. The mathematical calculation for the Dice coefficient similarity between documents $D1$ and $D2$ is calculated as:

$$Sim(D1,D2) = \frac{2\sum_{i=1}^{n} D1_i \cdot D2_i}{\sum_{i=1}^{n} D1_i^2 + \sum_{i=1}^{n} D2_i^2}. \qquad (3.4)$$

3.2.3 Cosine Coefficient

The Cosine coefficient is another method of computing vector similarity that has gained popularity over the years. It also generates similarity values that are within the [0,1] range. For two document vectors $D1$ and $D2$ it can be computed as:

$$Sim(D1,D2) = \frac{\sum_{i=1}^{n} D1_i \cdot D2_i}{\sqrt{\sum_{i=1}^{n} D1_i^2 + \sum_{i=1}^{n} D2_i^2}}. \qquad (3.5)$$

3.3 Document Classification

Classification is an act or process of systematic arrangement in groups or categories according to established criteria (Merriam-Webster Dictionary). Though it is recognized with diversified applications, the purpose of classification can be generalized as simplification and prediction in a large data collection [17]. In our particular case we are solely interested in applying the concepts of classification to automatically categorize a large collection of journal articles into a number of groups according to their content similarity. This is where the document similarity matrices discussed in the previous sections comes into the problem scenario. This act of classification that will be discussed in the next sections is intended to automatically provide information about the core research areas and innovations in the articles, the different dominant collaboration groups and the trend in which information is flowing in a particular field of interest.

3.3.1 Hierarchical Clustering

Hierarchical clustering is a type of clustering method that is popularly used in information retrieval systems. It produces a nested structure of partitions in a dataset based on a particular partitioning or merging criterion [26]. It includes two types of procedures namely divisive and agglomerative that process the clustering top-down and bottom-up respectively. An agglomerative hierarchical clustering starts by treating each data point as a separate cluster and merges the clusters that are the closest [13]. This process is repeated until a minimum number of clusters is achieved. Divisive hierarchical clustering starts with one cluster containing all the data points and divides them into sub clusters based on a criterion.

The distance between clusters r and s can be measured using different linkage methods [27].

- *Single linkage method:* this method takes the minimal distance between any two data points belonging to two different clusters as the distance measure between the clusters. Mathematically,

$$d(r,s) = \min(dist(x_{ri}, x_{sj})), i \in (1,\dots,n_r), j \in (1,\dots,n_s)\,. \tag{3.6}$$

- *Complete linkage method:* this method uses the maximum distance as opposed to the previous method. Mathematically,

$$d(r,s) = \max(dist(x_{ri}, x_{sj})), i \in (1,\dots,n_r), j \in (1,\dots,n_s)\,. \tag{3.7}$$

- *Average linkage method:* this method takes the average distance of all possible combinations of pairs of elements in the two clusters of interest as follows:

$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj})\,. \tag{3.8}$$

- *Centroid linkage method:* this method uses the distance between the centroids of the clusters.

$$d(r,s) = d(\bar{x}_r, \bar{x}_s)\,, \tag{3.9}$$

where $\bar{x}_r$ and $\bar{x}_s$ are the centroids of the two clusters.

- *Ward's linkage method:* also known as minimum variance linkage, uses the total sum of square distance within groups represented as:

$$d(r,s) = \frac{n_r n_s d_{rs}^2}{n_r + n_s}, \tag{3.10}$$

where d_{rs}^2 is the centroid distance between clusters r and s.

Lets consider the example of a two-dimensional data set shown in Figure 3.3 below.

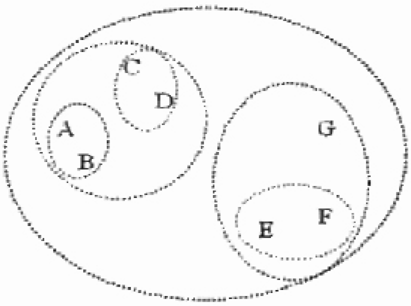


Figure 3.3 Example showing a two-dimensional data set

Application of agglomerative hierarchical clustering to this data set using a single linkage method would result in the tree-like structure shown in Figure 3.4. It starts by treating each of the seven data points as independent clusters. It then merges clusters that are closest to each other which in this case are data points A and B. Next C and D are grouped together followed by merging first group formed with this one. This process is repeated until a single cluster is formed. Except for the Ward's linkage method, the where $\bar{x}_r$ and $\bar{x}_s$ are the centroids of the two clusters.

- *Ward's linkage method:* also known as minimum variance linkage, uses the total sum of square distance within groups represented as:

$$d(r,s) = \frac{n_r n_s d_{rs}^2}{n_r + n_s}, \tag{3.10}$$

where d_{rs}^2 is the centroid distance between clusters r and s.

Lets consider the example of a two-dimensional data set shown in Figure 3.3 below.

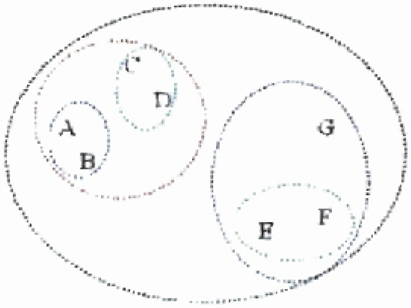


Figure 3.3 Example showing a two-dimensional data set

Application of agglomerative hierarchical clustering to this data set using a single linkage method would result in the tree-like structure shown in Figure 3.4. It starts by treating each of the seven data points as independent clusters. It then merges clusters that are closest to each other which in this case are data points A and B. Next C and D are grouped together followed by merging first group formed with this one. This process is repeated until a single cluster is formed. Except for the Ward's linkage method, the height of each branch in the tree representation shown in Figure 3.4 tells the distance between each of the different data points being connected.

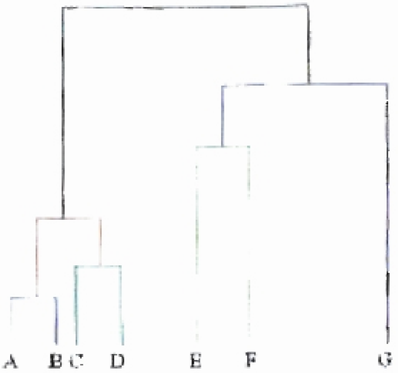


Figure 3.4 Result of agglomerative hierarchical clustering on the dataset shown in Figure 3.2

3.3.2 Fuzzy Based Clustering

In contrast to hierarchical clustering methods, fuzzy clustering does not produce hard disjoint partitions. Instead it uses a membership function to assign each data set to different clusters [26]. This helps overcome the problem of over-simplification imposed by other clustering algorithms in cases where an item belonging to more than one cluster has to be classified [17]. Figure 3.5 below shows a comparison between fuzzy clustering and hard partition clustering. Among the different types of fuzzy clustering methods, fuzzy c-means (FCM) clustering is the most popular one. FCM starts by placing centers for clusters inaccurately. It then moves these centers to minimize an objective function and updates the membership value for each data item simultaneously. The final output of this is a set of fuzzy cluster centers and membership values for each data point [28].

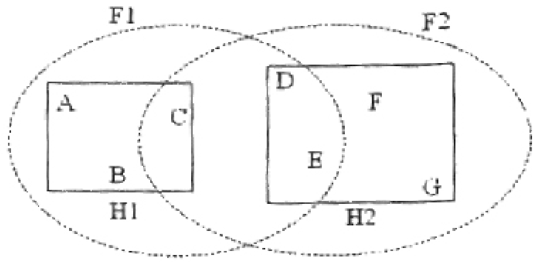


Figure 3.5 Example showing fuzzy and hard-partition clustering

The elements enclosed in the rectangles $H1 = \{A, B, C\}$ and $H2 = \{D, E, F, G\}$ are assigned to the groups with full confidence. But when it comes to the fuzzy clusters $F1$ and $F2$ shown by the ellipses, another parameter, namely membership value, is introduced. Thus an example membership description would be $F1 = \{(A,0.5),(B,0.8),(C,0.9),(D,0.2),(E,0.6)\}$ and $F2 = \{(C,0.1),(D,0.2),(E,0.7),(F,0.9),(G,0.6)\}$. The pairs (m, v) in each cluster represent the members and corresponding membership values. Thresholding the membership values can be used to form hard clustering result [26].

As mentioned above, fuzzy clustering gives a more practical result in cases where there are items that have both similarity and dissimilarity and need to be assigned to different clusters. This is widely used in areas of data analysis, patter recognition and image segmentation.

3.3.3 Neural Network Based Clustering

Artificial neural networks (ANNs), motivated by biological counterparts, have received a wide variety of applications in areas of science, engineering, mathematics, medicine, business, finance, etc. [29]. They are also used in specific applications for pattern recognition and classification purposes in such a way that they can be employed to help construct decision boundaries that can classify data sets from simple one-dimensional line to high-dimensional boundaries that are hard to visualize. A typical neural network architecture that can be used for classification is shown in Figure 3.6 below.

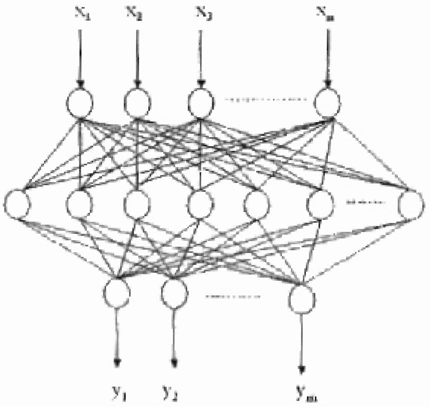


Figure 3.6 Example of classifier neural network

Each circle in the figure represents a neuron that is connected to others. The lines represent connections, which are assigned numeric values called weights. This neural network has *n*-dimensional inputs and *m*-dimensional output. It can be trained using different training method so as to tune its weighting parameters to be able classify a particular data set. After training it can be used to classify an *n*-dimensional data into one among a set of *m* groups.

Among the most commonly used ANNs in pattern recognition and classification self-organizing maps (SOM) [30] have attained a wide variety of applications for information retrieval and data mining applications especially in organizing [24] and visualizing [31] large and high-dimensional data collections [19]. It is constructed of a number of nodes arranged usually in a two-dimensional grid structure. These nodes later form groups upon training by moving around to preserve the topology of the input data structure [32]. After training the distance between data points directly represent their similarity, i.e. most similar ones appear close to each other while dissimilar ones are placed apart from each other. Figure 3.7 shows an example of a two-dimensional SOM structure.

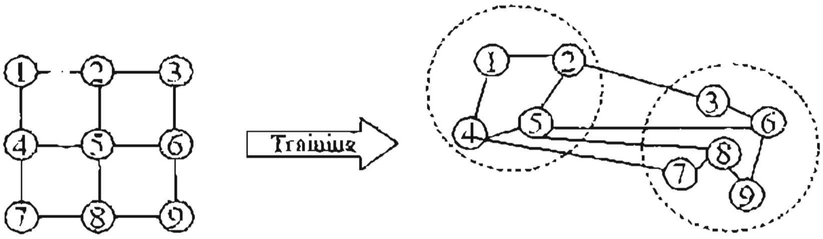


Figure 3.7 Training example of a 3×3 SOM structure resulting in two distinct groups

As can be seen in the above figure, training of a 3×3 SOM produced two distinct groups with which incoming data will later be classified with based on how similar it is to either of the groups. The map could also convey graphical representation of a high-dimensional input data.

CHAPTER IV

INFORMATION FUSION

4.1 Overview of Information Fusion

Information fusion is a technique of using different information gathered from multiple sources such as databases, sensors, human collected data, etc. to get a better and more precise knowledge and understanding about a specific subject. Information fusion has been applied in a variety of applications such as image recognition, signal processing, sensor fusion, information retrieval, etc. In this chapter the issue of fusing different types of similarity information gathered from a collection of journal articles will be examined. A discussion is also given on the method proposed to fuse the different types of similarity matrices developed in the previous chapter so as to generate a better classification of the articles that will help understand the underlying subject better and explore it at a more detailed level.

The proposed method of information fusion is needed in order to be able to classify the collection of journal articles using the information gathered from the different similarity matrices. Each similarity matrix processes specific characteristic of information that it inherits. Similarity information gathered from bibliographic citation information displays the flow of knowledge and information within the collection. This is because each innovative research is built on a base knowledge from which it derives all of its assumptions and knowledge as a starting point.

In addition to bibliographic citation information, this research proposes that other similarity information gathered from the authorship and word content analysis could also be used to enhance the overall knowledge about the collection, if used appropriately. Similarity information extracted from authors' identity provides special information that can help to identify the different research collaboration groups within the collection. This information can be used to strengthen the similarity analysis between the articles that belong to the same or related research area. In addition, this will maintain and give a better understanding of the different social network of authors within the community.

In a similar way, word content analysis has also a special role that would contribute towards achieving a better classification. In this research a basic level word content analysis is carried out on the collection to extract as much useful information as possible. Words, in the context of similarity information extraction, are generally 'noisy' as there are several problems associated with them. These problems are faced particularly when converting them into a quantitative representation. For example, there is the problem of polysemy and synonymy mentioned earlier. This problem of a word having several meaning and many words having the same meaning is hard to quantify with out human intervention. There is also the problem of evolution of language, which is a major challenge when trying to bring documents on a wide range of time frame together because terms used in almost every language evolve through time. Even so when dealing with research articles especially, most research-related terminologies convey particular and unambiguous information that can be used to classify the articles. The approach taken towards constructing the similarity information from word frequency analysis is described in detail in the following section.

4.2 Similarity Information Gathering

In this section the similarity information gathering process of the journal articles will be discussed in detail. The scope of this research is focused only on similarity information extracted from bibliographic citations, author information and word content analysis.

4.2.1 Bibliographic Citation Similarity

Given a collection of n documents and m references, an $n \times m$ paper-reference representation matrix PR can be formed, where P stands for paper and R for references. Here usually m tends to be much larger than n because a paper commonly cites more than one reference and different papers have different reference lists. An element of the PR matrix, $PR(i,j)$, is set to one if reference j is cited in paper i. As a result, this matrix is normally a sparse matrix with most of its entities having value of zero.

Having this PR matrix, the citation similarity information can be calculated using the dice coefficient discussed in the previous chapter as follows,

$$S_r(i,j) = \frac{2 \times C_r(i,j)}{N_r(i) + N_r(j)}, \tag{4.1}$$

where $S_r(i,j)$ is the citation similarity between documents i and j, $N_r(i)$ is the number of total references in document i, and C_r is a reference co-occurrence matrix which can be calculated as:

$$C_r = PR \times PR^T. \tag{4.2}$$

The value of $C_r(i,j)$ indicates the total number of common references between documents i and j.

4.2.2 Author Similarity Information

In a similar fashion, the author similarity matrix can be computed as follows,

$$S_a(i,j) = \frac{2 \times C_a(i,j)}{N_a(i) + N_a(j)}, \tag{4.3}$$

where $S_a(i,j)$ is the author similarity between documents i and j, $N_a(i)$ is the number of total authors in document i, and C_a is an author co-occurrence matrix which can be calculated as:

$$C_a = PA \times PA^T, \tag{4.4}$$

where PA refers to the paper-author matrix defined in the same way as the PR matrix.

4.2.3 Term Similarity Information

The other similarity matrix constructed for the collection of the articles is a term similarity matrix. The steps taken towards the construction of this matrix are as follows. First each word in the abstract of every article was parsed and entered into a database excluding a list of user-specified stop-words that did not bear with any particular meaning. A basic word processing was also performed on the parsed words so as to avoid different versions of same word by removing common prefix and suffixes such as *re*, *ing*, *ous*, etc. After this, the top t most frequent terms were selected as representing features for the document collection. This value of the threshold was set depending on the total number of words extracted and size of document collection. Next an $n \times t$ paper-term information matrix PT that contained the frequency or number of occurrence of each term in each document was constructed. $PT(i, j) = b$ implies that paper i contains term j b number of times.

Next the same approach as the previous ones was taken to calculate the term similarity matrix of the entire document collection as follows.

$$S_t(i, j) = \frac{2 \times C_t(i, j)}{N_t(i) + N_t(j)}, \tag{4.5}$$

where $S_t(i, j)$ is the term similarity between documents i and j, $N_t(i)$ is the number of selected terms in document i, and C_t is a term co-occurrence matrix which can be calculated as:

$$C_t = PT \times PT^T. \tag{4.6}$$

4.3 Similarity Information Fusion

After the above three types of similarity information matrices were derived a weighted sum scheme was used to fuse them and form a single composite similarity matrix. The weighting was done as shown in Equation 4.7 below.

$$S_f = w_r \cdot S_r + w_a \cdot S_a + w_t \cdot S_t , \qquad (4.7)$$

where S_f represents the final similarity matrix and w_r, w_a and w_t are weighting coefficients that satisfy the equation:

$$w_r + w_a + w_t = 1 , \qquad (4.8)$$

where $w_r, w_a, w_t \in [0,1]$.

These weighting coefficients should satisfy Equation 4.8 because the similarity values calculated in the previous section are always between zero and one, where a zero value implies no similarity at all and a similarity value of one represents total similarity. Hence, the final similarity matrix S_f formed using Equation 4.7 is also made to satisfy this condition.

The optimal choice of these weighting coefficients is derived using an evolutionary genetic algorithm based search. The input space for these coefficients can be schematically shown as in Figure 4.1 and every point lying in the surface is a possible candidate for the best weighting coefficients. This presents an infinite number of candidates for the weighting coefficients.

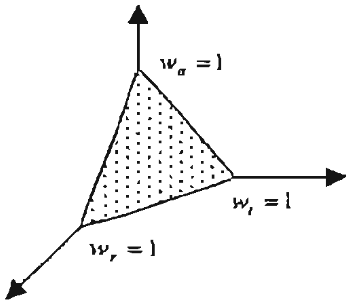


Figure 4.1 Input space of the weighting coefficients

4.4 Genetic Algorithm Based Search

4.4.1 Overview of Genetic Algorithms

Genetic algorithms (GAs) are population based point-by-point search algorithms that can be used to solve different types of search and optimization problems [13]. In analogous way to natural genes, different characters of population members are encoded within binary bits of strings containing zeros and ones. Different genetic operations such as reproduction, crossover, and mutation are performed on these genes through time [33, 34]. The survival of the fittest principle applies at every generation and only those population members that perform well are most likely to survive and give offspring that share their qualities. GAs are different from traditional search algorithms in such a way that they are not deterministic; rather they are stochastic in nature. They also perform search from a population rather than just a single possibility [33].

Summary of the GA search algorithm is given in the table below.

Table 4.1 Process of a typical Genetic Algorithm Process

```
1. Set iteration index $i$ to 0.
2. Generate $P(i)$ number of populations at random.
3. REPEAT
      a. Evaluate the fitness of each individual in $P(i)$.
      b. Select parents from $P(i)$ based on a fitness criterion function.
      c. Produce next generation $P(i+1)$ using genetic operations.
      d. Set $i=i+1$.
   UNTIL the stopping criterion is met.
```

4.4.2 Why Genetic Algorithm for Weighting Coefficient Search?

A GA based search was chosen to search for the optimal weighting coefficients for two reasons. One reason is that given an infinite number of possible solutions, GA can do a better job in finding the best candidate with a fairly less computational complexity. Another reason for choosing GA is to make the text classification architecture scalable to cases in which there are more than three similarity information to be fused together. Imagine performing a direct point-by-point search on a high dimensional space, which is not practically a recommended idea. Instead, GA can be used to efficiently search for the best weighting coefficients even when the number of dimensions increases.

4.4.3 Genetic Coding and Search for the Weighting Coefficients

GA encodes any candidate solution to a problem as a gene in terms of bits of zeros and ones. A collection of this kind of genes makes up an entire population that will be used to search for the best solution. In this particular problem of search for the best three weighting coefficients, the problem can be scaled down to search for two weighting coefficients since the third one can be found by using Equation 4.8. This is shown in Figure 4.2 below.

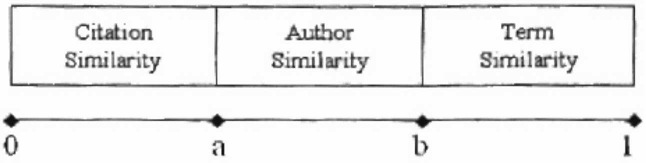


Figure 4.2 Two dimensional version of the weighting coefficients' search problem

Now the problem is only about getting the two parameters *a* and *b* and the coefficients can be calculated as:

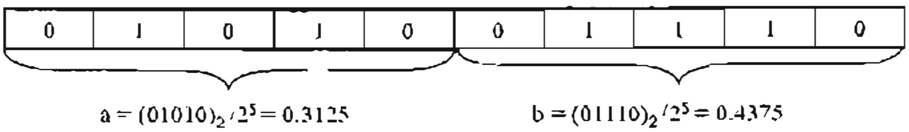
$w_r = a$, $w_a = b - a$, and $w_t = 1 - b$.

The table below gives some examples that describe the relationship between the values of *a* and *b* and the composition of the final similarity matrix.

Table 4.2 Practical examples on the weighting coefficients

a	b	Meaning – Final Similarity Matrix Composition
0.0	0.0	100% Term Similarity
0.0	1.0	100% Author Similarity
1.0	1.0	100% Citation Similarity
0.1	0.5	10% Citation, $(0.5-0.1)\times 100\% = 40\%$ Author, $(1-0.5)\times 100\% = 50\%$ Term Similarity

With the above type of representation, the genetic coding for the parameters a and b can be done in the following way. Let each chromosome contain $2n$ number of genes of which the first n genes represent a and the rest b. The actual value of a or b can be calculated as the binary equivalent of the genetic sequence divided by 2^n which will result in a number between zero and one. The value of n is set to meet the desired level of resolution, i.e. level of increment between search parameters. This representation is exemplified in Figure 4.3 shown below for $n = 5$.



0	1	0	1	0	0	1	1	1	0

Figure 4.3 Genetic coding example of the weighting parameters

With this representation, genetic operations including reproduction, mutation, and crossover can be performed on the two parts separately to produce offspring and undergo normal genetic processes. The fitness function for every member of the population is evaluated after obtaining the clustering result for it and using the clustering performance evaluation function described in the next section. After a number of generations a stopping criterion, normally maximum number of generations or desired amount of fitness value whichever comes first, is reached and the values of the chromosome with best fitness value are taken as the final similarity information weighting coefficients. These coefficients are used to construct the final composite similarity matrix and an agglomerative hierarchical clustering is performed to obtain the final clusters using Ward's linkage method [1]. After this, timeline visualization [2] and interpretation of the data proceeds.

4.5 Clustering Performance Evaluation

As in any GA routine, a well-designed fitness evaluation function for each individual in the population is essential to search for the best weighting coefficient. In this research the following two possible methods were developed in order to evaluate clustering performance for the journal article classification.

4.5.1 Pareto Distribution Coefficient

Scatter within each final cluster was modeled as a Discrete Pareto Distribution [35] and the model exponent (γ) was used as a measure of scatter. In a Pareto Distribution, small occurrences are very common and large ones very rare. Figure 4.4 exemplifies this idea by plotting a log-log graph of frequency (*f*) versus number of papers referenced *f* times.

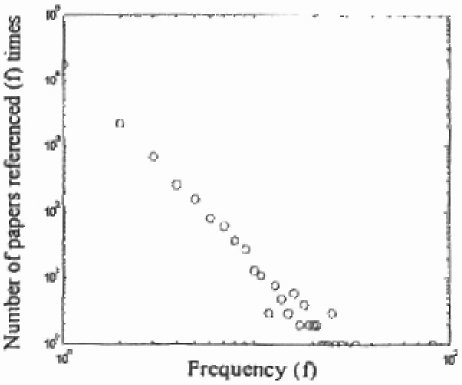


Figure 4.4 Log-log plot showing the characters of a Pareto-Distributed citation frequency data

As can see in the figure, in the example collection of papers considered there are only few papers that were cited high number of times and many papers, plotted towards the tail of the graph, that were cited only few number of times.

The Pareto Distribution coefficient (γ) is the slope of the linear curve fit through the data. An increase in the value of this coefficient indicates the minimization of scatter in the collection. A large value of γ would on the other hand indicate high degree of scatter and the goal here is to minimize the amount of scatter in each cluster. In other words, each cluster needs to be as specific to a particular research topic as possible. This would result in a smaller coefficient γ.

The following example explains this idea of measuring scatter within a collection in terms of γ. Figure 4.5 shows the collection of 833 documents clustered into 10 groups and plotted as dots with their publication dates as x axis and cluster membership as y axis. The dark dots are documents being selected for example purpose.

First 30 documents were selected at random and γ was computed for the key terms in the documents and a value of $\gamma = 2.75$ was obtained. Next the same number of documents but now all belonging to the same cluster was selected. This time a lower value $\gamma = 2.3$ was obtained. The document samples taken and their Pareto Distribution curve are shown in Figures 4.5 and 4.6.

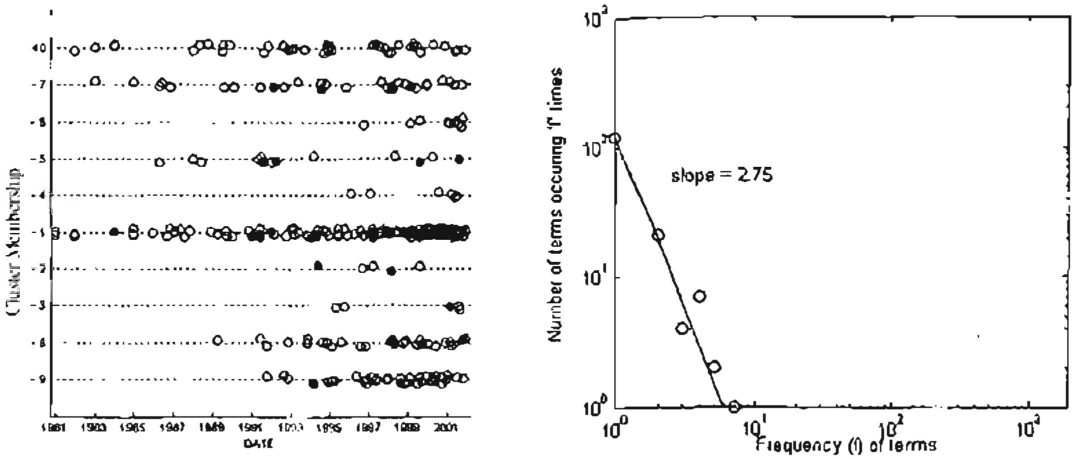


Figure 4.5 Documents belonging to different cluster chosen at random selected (left) and the Pareto distribution of their key terms (right)

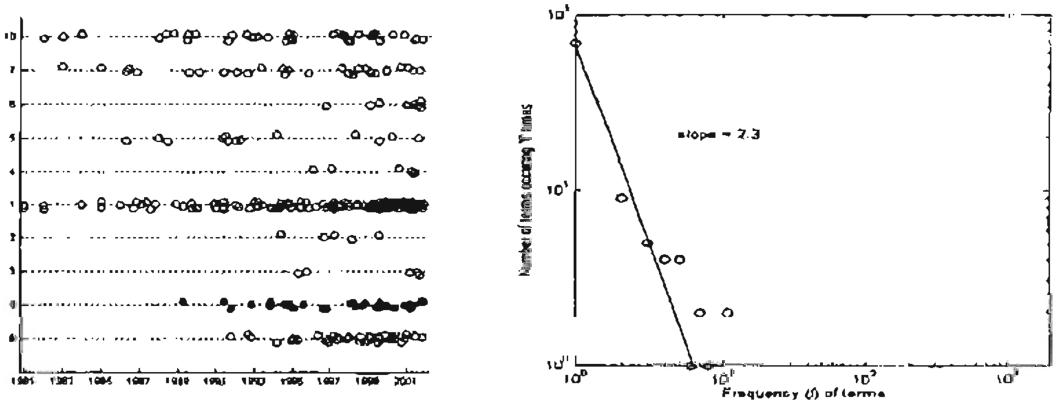


Figure 4.6 Documents belonging to the same cluster selected (left) and the Pareto distribution of their key terms (right)

This experiment was repeated a number of times and all results showed that random selection had a greater value of γ than selection of documents belonging to a particular group. This criterion was use as a means of evaluating the similarity information fusion technique used.

4.5.2 Linkage Between and Within Clusters

Another classification performance evaluation that was considered was minimization of citation linkage between clusters and maximization of linkage within clusters. A high number of average linkages within each individual clusters has the direct implication that the final clusters formed have provided strong connections within themselves, which is a direct indication that all the articles that are closely related have been categorized accordingly. Minimization of average number of citation linkages across clusters also indicates that we have managed to form a clear-cut grouping by making the classification as definite as possible. This idea is illustrated in Figures 4.7 and 4.8 below.

In this example, seven documents, represented by circles with a number on them and lines as a citation link, are being clustered into two groups. The clustering result in Figure 4.7 shows several links that cross over from one cluster to another. However, a closer observation would reveal that if document 6 moves to group 1 and document 1 to group 2, much of the cross over links would be removed. In Figure 4.8 the same documents are being re-clustered to produce a better cluster that presents the minimal number of links across the two groups.

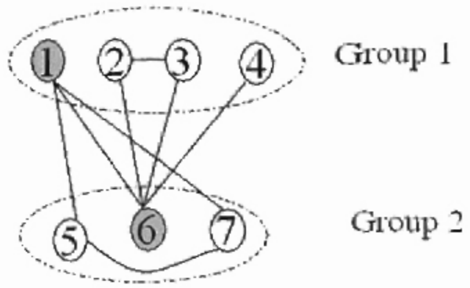


Figure 4.7 Example of poor clustering

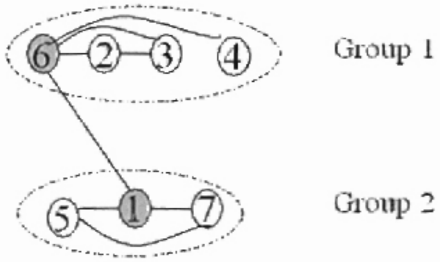


Figure 4.8 Clustering with a better performance

4.6 Experimental Results on Information Fusion

The clustering performance evaluation techniques proposed above were used to evaluate the idea of similarity information fusion on different types and proportion of similarity matrix composition. The tests simulations performed showed that similarity information fusion helps the clustering routine do a better job. The simulation results for these tests are discussed below.

The first test performed was citation and author similarity fusion to evaluate the clustering performance with respect to the average Pareto Distribution coefficient (γ) of the key (index) terms within the journals in every cluster formed. The fusion in this test was done according to Equation 4.9.

$$S_f = w \times S_r + (1 - w) \times S_a , \tag{4.9}$$

where w is a weighting coefficient that was varied from zero to one to test for different compositions. The result of this test is shown in Figure 4.9 below.

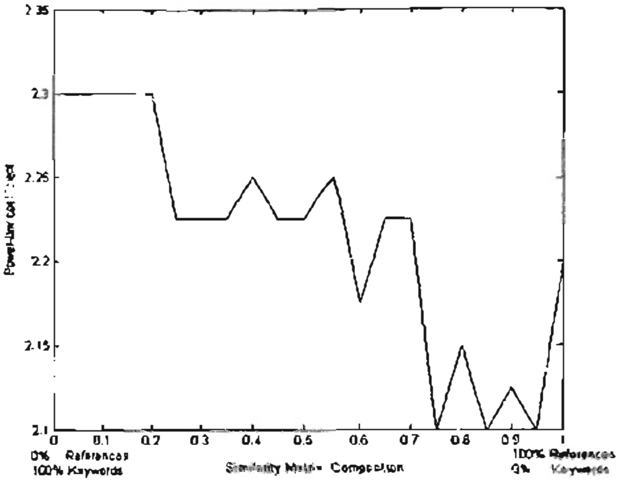


Figure 4.9 Plot showing the reduction of the average Pareto Distribution coefficient for key terms by fusing small amount of author information

This plot shows the average value of γ for all the clusters formed at every value of w and from the result we can see that minimal values of γ are achieved for $w = 0.75$, $w = 0.85$ and $w = 0.95$ for this particular experiment. Minimization of γ with respect to index terms indicates that the final clusters formed based on the fused similarity matrix had many of those articles with similar index terms clustered together which would reduce the slope of the linear-curve-fit shown in Figure 4.4 by including more high frequency terms towards the tail of the plot.

Similar experiment was also performed to show the reduction of γ for cited references. This time γ was calculated for the references of the articles within each cluster. The table below shows an example of cited-reference frequency extract of one cluster.

Table 4.3 Example showing cited reference frequency count

Citation	Frequency
LEPPLA SH, 1982, P NATL ACAD SCI USA, V79, P3162	9
DUESBERY NS, 1998, SCIENCE, V280, P734	9
INGLESBY TV, 1999, JAMA-J AM MED ASSOC, V281, P1735	8
VITALE G, 1998, BIOCHEM BIOPH RES CO, V248, P706	7
PETOSA C, 1997, NATURE, V385, P833	6
KLIMPEL KR, 1994, MOL MICROBIOL, V13, P1093	6
MILNE JC, 1993, MOL MICROBIOL, V10, P647	4
FRANZ DR, 1997, JAMA-J AM MED ASSOC, V278, P399	4
FRIEDLANDER AM, 1986, J BIOL CHEM, V261, P7123	4
MILNE JC, 1994, J BIOL CHEM, V269, P20607	4
MESELSON M, 1994, SCIENCE, V266, P1202	4
LEPPLA SH, 1988, METHOD ENZYMOL, V165, P103	3
HENDERSON DA, 1999, SCIENCE, V283, P1279	3
MILNE JC, 1995, MOL MICROBIOL, V15, P661	3
...	...

Table 4.3 shows frequency of the top cited references within a single cluster. Minimization of γ in this case would imply that the similarity matrix composition was able to bring together the key references into single clusters. This would help understand the final clusters better, as it would collect the references serving as a knowledge base together. Same technique as the previous one was performed to explore the performance of the clustering by fusing citation and author information and the simulation result for this test is shown in Figure 4.10.

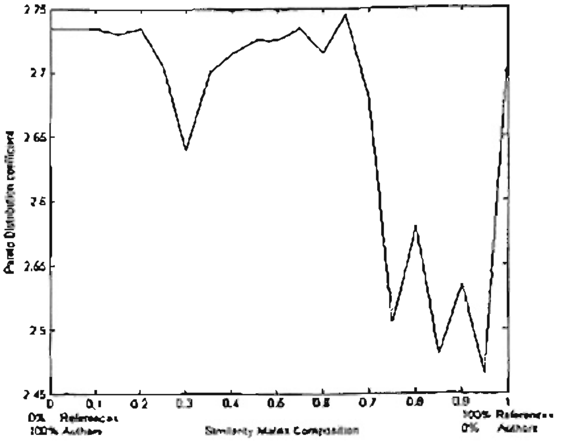


Figure 4.10 Plot showing reduction of the average Pareto Distribution coefficient for cited references by fusing small amount of author information

4.7 The Final Picture

4.7.1 Algorithm Summary

In this section all the ideas that have been discussed so far are put together into one complete algorithm proposed to transform a collection of journal articles into a more meaningful presentation of information that is expected to provide answer for the following questions and more.

- What are the main research topics within the collection?
- Who are the experts in these areas?
- What collaboration groups are there?
- When and what are the major discoveries in the past?
- Where is the technology going?
- Etc.

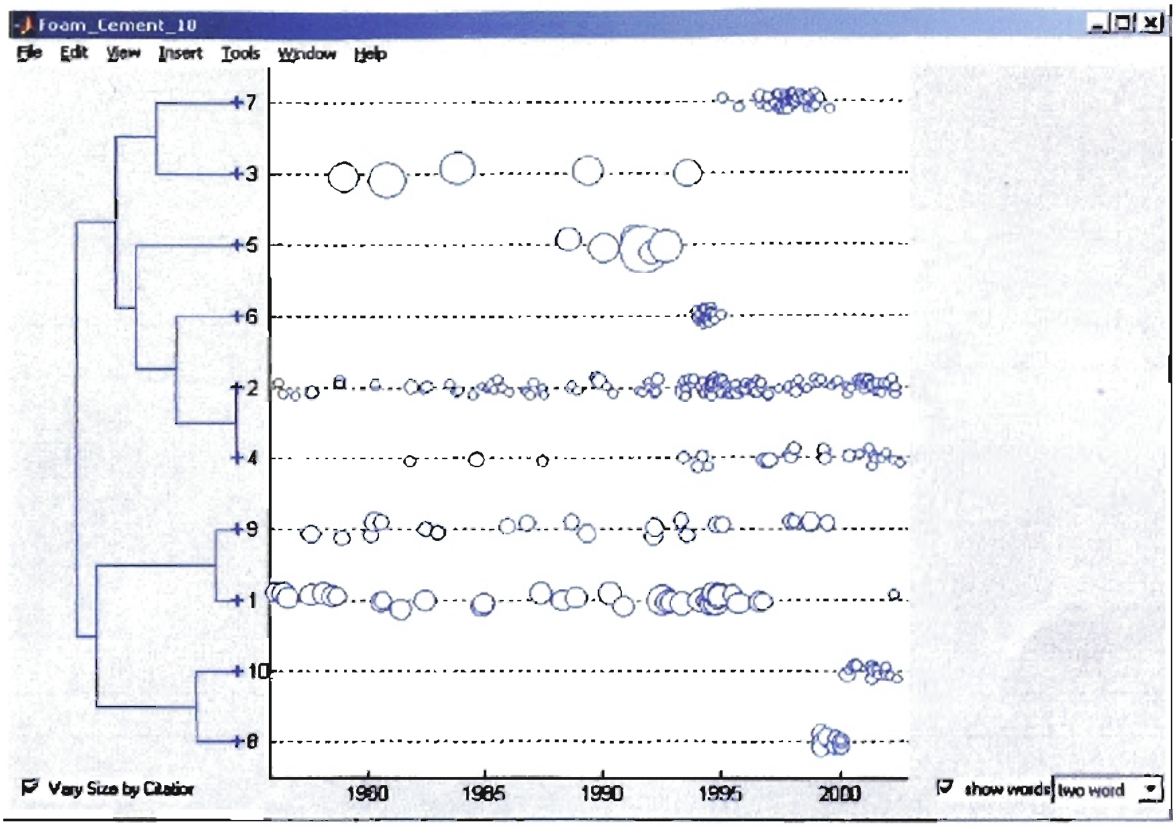
Table 4.4 Outline of the text classification algorithm

TEXT CLASSIFICATION ALGORITHM

1. Collect data.
2. Pre process the data.
    - Includes parsing, removing redundancies, extracting terms from titles and abstracts.
3. Extract different similarity information.
    - Construct similarity matrices based on all available sources including term analysis, citation analysis, author analysis, etc.
4. Perform a Genetic Algorithm based search for the best similarity matrix weighting coefficients and fuse the similarity matrix using the best coefficients.
    - Determine the genetic encoding, genetic operations, population size and stopping criterion.
    - Define the fitness function to be used depending on the type of the collection and desired characteristics of the classification result.
5. Perform classification of the collection based on the final similarity matrix formed.
    - Use the type of linkage function (single, centroid, Ward, etc.) and classification (agglomerative, divisive, or other) that best suits the application.
6. Visualize and interpret the result.
    - If temporal information is available show the results as time lines and do further exploration on the final result.

4.7.2 Visualization and Interpretation

After classification is performed based on the fused similarity matrix, the next step is to visualize the result and interpret it. The method of visualization that is used in this research is a time line visualization technique [2] discussed below. The DIVA software tool [19] was also used to explore the result. The Figure 4.11 below shows a case study done on a collection of patents on the topic of petroleum oil well foam cements extracted from the US Patents Database. The study consists of 333 US patents.

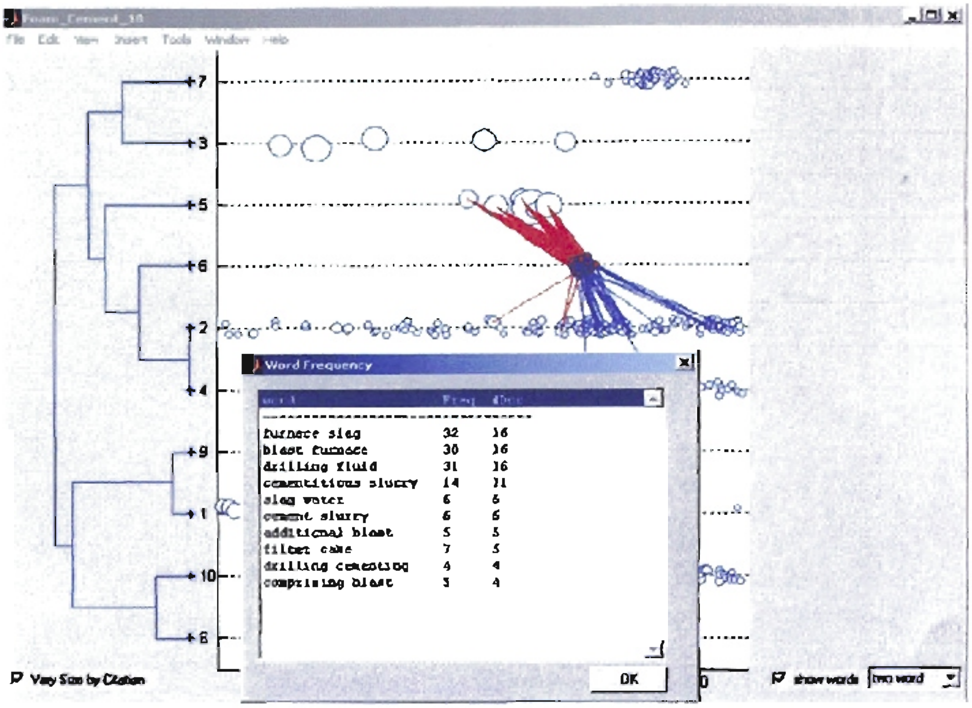


☑ Vary Size by Citation

☑ show words two word

Figure 4.11Time line visualization of a case study on foam cements

This graph conveys the following information. The tree structure on the left side of the figure shows the structure of the hierarchical classification. The dots on the map represent documents (patents in this case) and their x-axis corresponds to their date of publication and y-axis corresponds to the cluster to which they belong. The size of the documents can be made to vary according to the number of times they were cited by checking the box on the bottom left of the window. Thus the documents with higher number of citation hits appear as large dots and can be further studies in a more detail. The area on the left side of the figure is left for labeling, which is done manually by studying and exploring every cluster. This can be done by selecting documents and exploring the content of their word frequency as shown next.



```
word                     Freq  #Doc
-----------------------------------
furnace slag              32    16
blast furnace             30    16
drilling fluid            31    16
cementitious slurry       14    11
slag water                6     6
cement slurry             6     6
additional blast          5     5
filter cake               7     5
drilling cementing        4     4
comprising blast          3     4
```

Figure 4.12 Exploration process of the result

Selecting the *show words* option on the right bottom and dragging a rectangle on documents of interest will popup the word frequency window shown on top of the time line result and the user can judge what the cluster or group is about by studying the frequency of the words within the selection. The lines on the graph represent citation links to and from the selected documents, which appear as dark dots. The final labeled map is shown in the next figure.

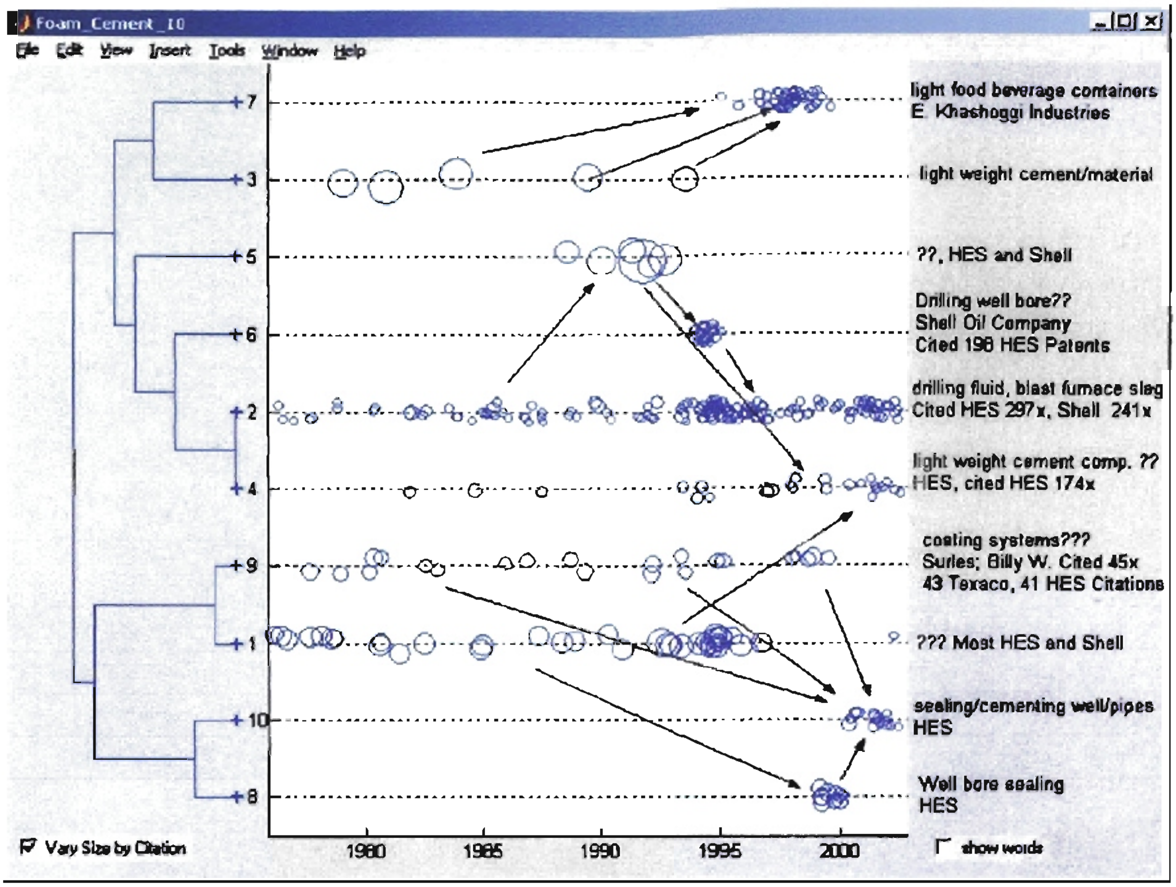


Figure 4.13 Final labeled map

This map shows all the cluster titles that are the result of the exploration. The arrows are also drawn to show the trend of citation through time across the different groups that help to give an idea on the direction in which information is flowing. The clusters with large number of documents can also be further classified by clicking on the cross sign at the legs of the tree structure.

4.7.3 Simulated Annealing for Optimized Visualization

The proposed method uses a modified version of the time visualization in order to avoid misconceptions about the final clusters. The original time line visualization technique presents the results in the order they were generated by the hierarchical clustering routine. However, clusters plotted close to each other are not necessarily similar. The method of optimization introduced in this research uses simulated annealing [36] based flipping of branches of the tree structure to come up with an ordering in which the most similar clusters appear close to each other without altering the tree's structural information. A diagrammatical example for this is given in Figure 4.14.

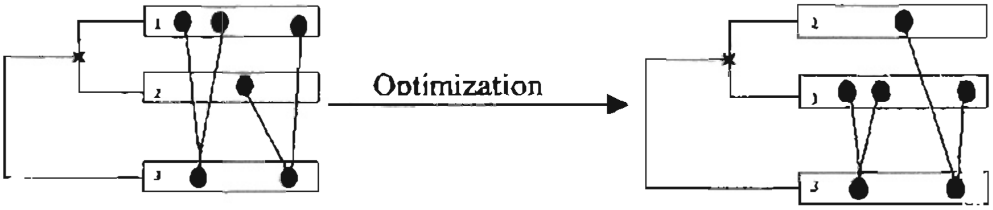


Figure 4.14 A simulated annealing based optimization for the time line display

In this example, the node marked with an "X" mark is flipped without changing the tree structure. However this change has made the most similar clusters, 1 and 3, that have high number of connections to appear close to each other. This method help to better understand and interpret the final result. An example on the improvement of visualization using this method is given in the next chapter.

CHAPTER V

CASE STUDY ON ANTHRAX

5.1 Information Collection and Preprocessing

In this chapter a case study conducted on the subject of anthrax using the technique developed in this thesis is presented. The study was performed based on a collection of articles obtained from the ISI Science Citation Index library using the query phrase "anthrax anthracis". This query returned articles published early from 1945 to the beginning of 2003. The summary of the documents obtained is given in Table 5.1 below. These articles were obtained in the form of a set of tagged text documents and were later parsed and stored into a Microsoft Access database. The procedures discussed earlier were then applied to classify the articles and develop a time line presentation of the collection. A starting population of 50, each with total number of bits equal to 15 was used in the genetic search algorithm. The fusion parameters obtained for this particular example dataset w_r, w_a and w_t were 0.78, 0.15, and 0.07 respectively.

Table 5.1 Summary of documents collected

Total number of	Count
Papers	2,472
References	25,007
Authors	4,493

Out of the 2,472 articles returned, only those articles that had 5 or more citation links to others were considered for further analysis. As a result, the number of articles under the study was reduced to 987. This helped exclude documents that were not of much relevance.

5.2 Presentation of the Results

After classification was performed on the collection of the articles based on similarity information extracted from the citation, author and terms, the result was plotted as a time line that was optimized for visualization using the simulated annealing routine introduced earlier in Chapter 4. The improvement of this routine on the display is shown in Figure 5.1 below. The green lines show similarity connection between the documents that was greater than a threshold value of 0.2. As can be seen in this figure, the simulated annealing routine changed the order in which the hierarchical the tree structure is organized which has resulted in a reduced number of crossover linkages between clusters. This optimization helps achieve better visualization while exploring the collection.

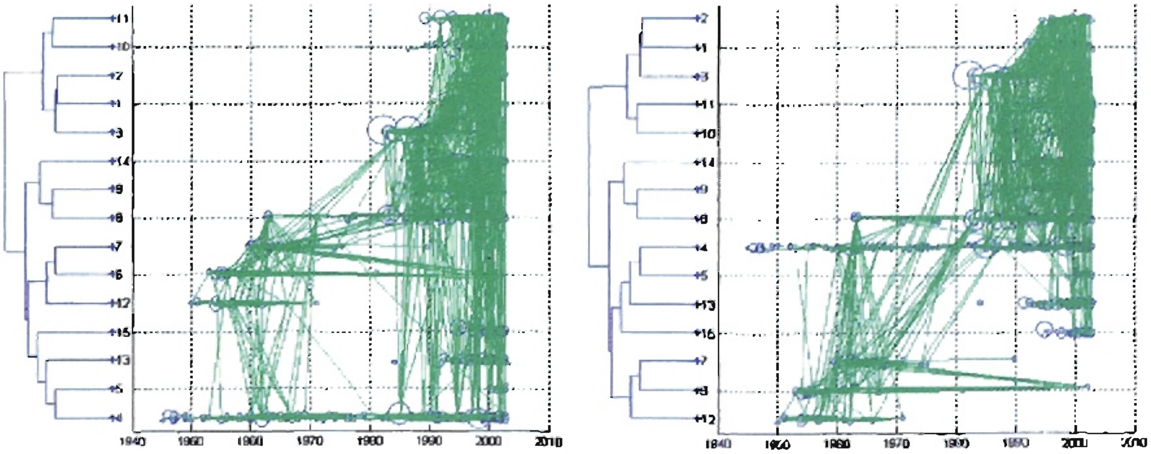


Figure 5. 1 Improvement of visualization before (left) and after using simulated annealing.(right)

The optimized time line result is shown with out any connections in Figure 5.2 below.

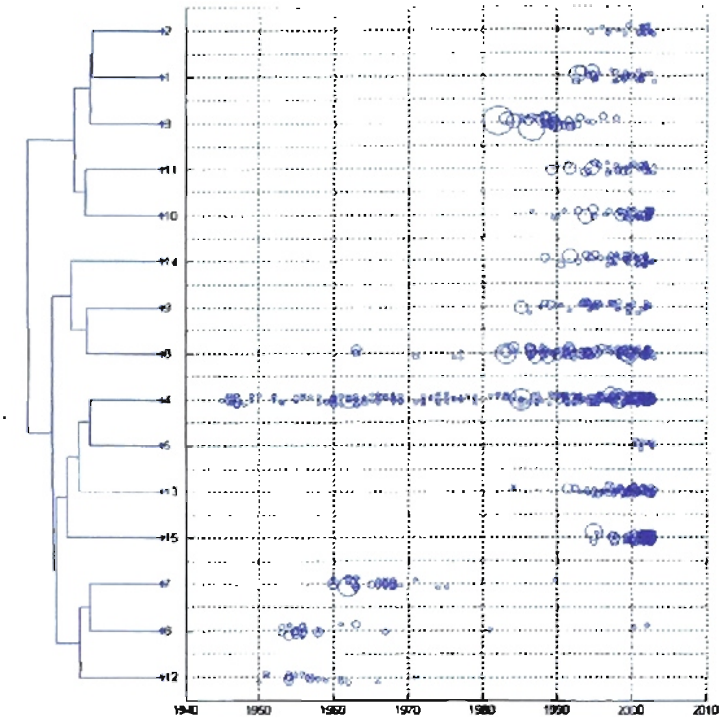


Figure 5.2 A first look at the classification

The time line shows the 987 documents plotted according to their publication date versus cluster membership corresponding to a particular research area. The relative size of each dot represents the number of times it was cited within the collection. This helps identify those documents that have been heavily cited by others graphically. The tree structure on the left side of the plot provides information about the structure of the clusters formed.

Figure 5.3 below shows a labeled version of the previous time line. The labels were made by taking a close note at the word frequency content of the articles' titles and abstracts within each cluster. The heavily cited articles are also marked with their topics and number of total articles citing them.

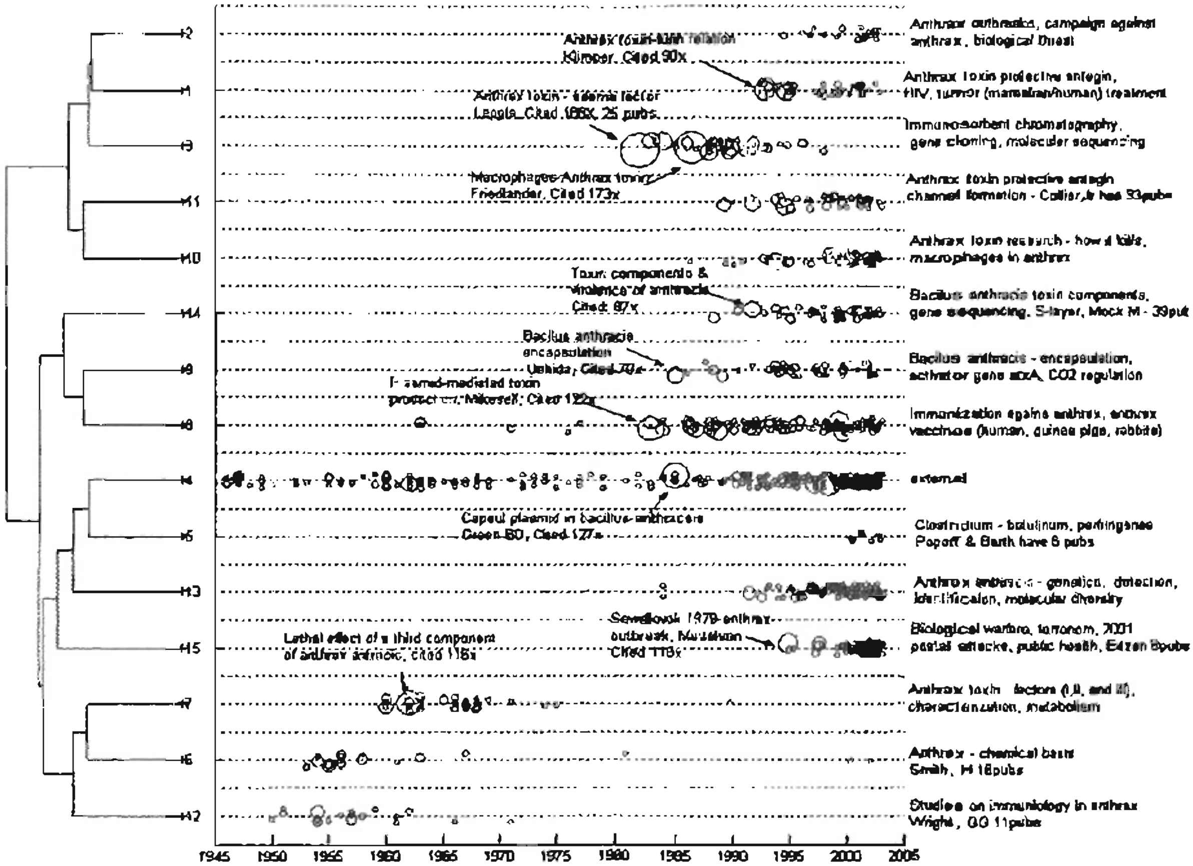


Figure 5.3 Labeled map of the result

This way of presentation can be used as a starting point for the analysis of the collection and study of the anthrax topic. This map reveals the different research areas related to anthrax research, experts and their main expertise, major findings in the field, time line information about the collection, and knowledge about the flow of information among the different research areas. This map can also lead to discoveries on emerging research areas and potential developments.

Clusters 7, 6, and 12 as shown in Figure 5.3 contain articles on "preliminary research" in anthrax mostly published between the 1950s and 1970s. As can be seen from the labels, these researches dealt with anthrax immunology and vaccines. These documents were later used by other documents for the new emerging researches as shown in Figure 5.4 below. The dark forward arrow in this figure shows the flow of information within the different research areas through time. The green lines show the strong connection between documents that have a similarity value greater than a threshold value of 0.3.

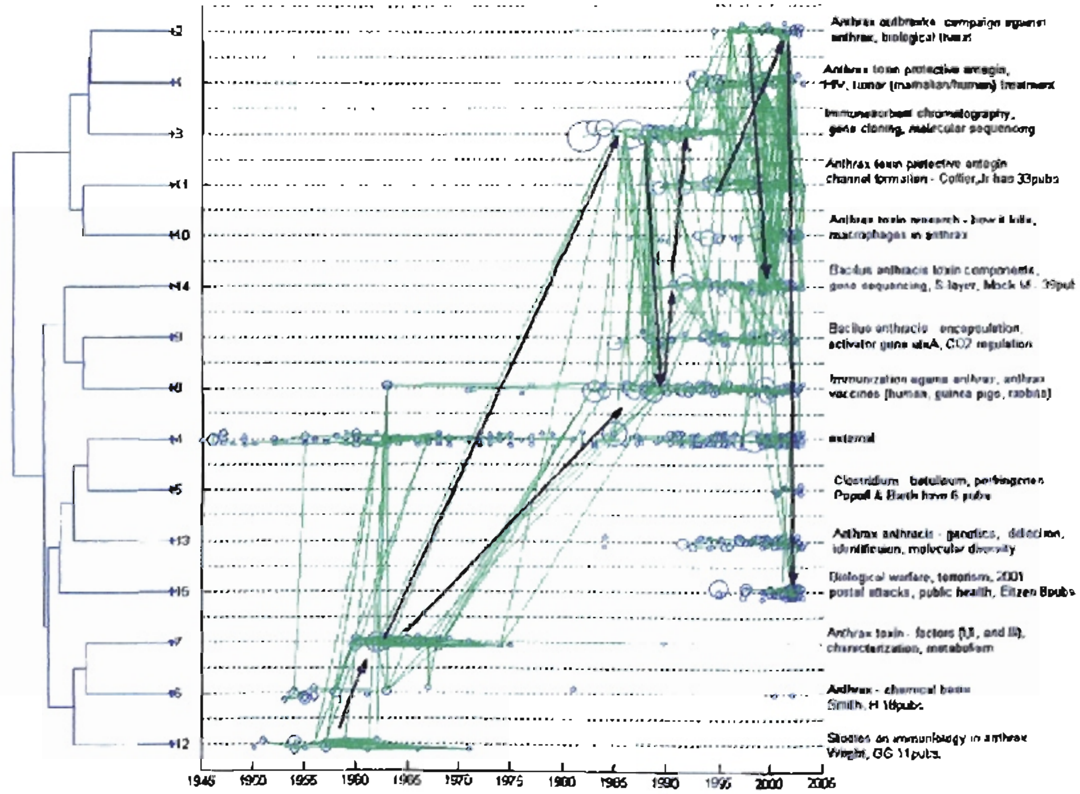


Figure 5.4 Figure showing the flow of information within the collection

From Figure 5.4 we see that the "preliminary research" articles served as base document for the emerging researches in gene cloning, molecular sequencing, anthrax toxin, and immunology. Base documents are defined as documents serving as a starting point for new emerging research topics. They are characterized by being heavily cited.

It is also worth noting that the cluster on bioterrorism, cluster 15, had its base documents from cluster 2, which contains articles reporting the different outbreaks around the world and biological threats of anthrax. Cluster 15 contains articles that were related to the US postal attacks, reports on inhalational and intestinal anthrax, and risks and prevention methods. Cluster 4, marked as "external", contains documents that did not have strong relation to any of the researches in collection.

The influence of the documents that had greater number of citations was also studied closely to identify the flow of information. As an example, the citation structure of the seminal article by Leppla in 1982 on edema factor, one part of the anthrax toxin that kills, was examined. This article has 188 citations and Figure 5.5 shows these citation links. This document is shown as a red dot in cluster 3. The red lines show papers cited by this article and the blue lines show the citations made to the article. As can be seen in this figure, the finding of Leppla presented in this article was used intensively on researches performed afterwards and we can conclude that it had a major contribution. We can also note that the finding in Leppla's article based on early anthrax research shown in clusters 6 and 7. These procedures can be followed to study on the different major contributions in the field.

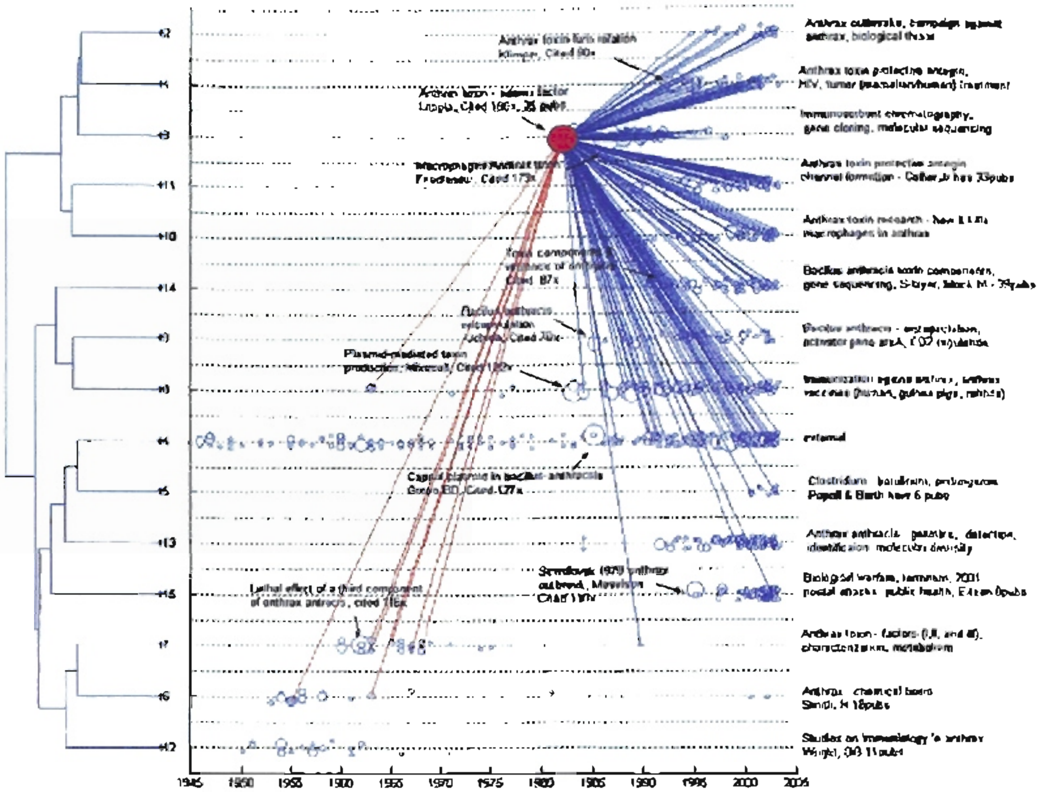


Figure 5.5 Citation structure of Leppla's article

The following example shows a summary of the collaborators of Leppla, who has 74 publications in this collection. Table 5.2 below gives a list of those people with whom Leppla published at least seven times.

Table 5.2 Major collaborators of Leppla

Author	Count
KLIMPEL, KR	16
SINGH, Y	12
ARORA, N	9
Liu, SH	8
LITTLE, SF	8
FRIEDLANDER, AM	7
GORDON, VM	7

The selected documents shown in red in Figure 5.6 below show the articles that Leppla published. From this map we can see that most of Leppla's publications were in the area of anthrax toxin and immunization.

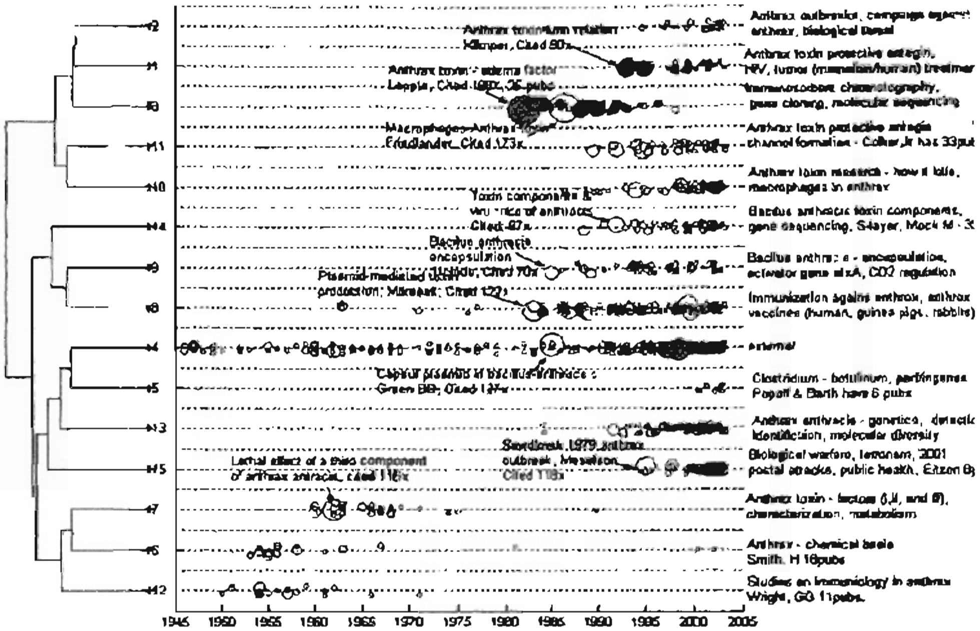


Figure 5.6 Articles published by Leppla

5.3 Summary

The study presented above identified the different research areas, major experts and their center of excellence, time information on the beginning and end of a particular research topic. This study can be done at different levels depending on the type and depth of information required from the article collection. From the results presented in the previous section, we can conclude that the method used to study the articles was able to identify the different research areas and classify the articles accordingly. The time line visualization technique was also a helpful tool in presenting and further exploring the result.

CHAPTER VI

CONCLUSION AND FUTURE WORK

6.1 Research Conclusions

This research involved automatic classification and categorization of collection scientific literatures into their corresponding research topics by using multiple similarity information extracted from their citation, author, and word content analysis. Each similarity matrix extracted from the collection emphasizes and contains information on different aspects of the collection. Classification based on similarity information extracted from citation information helps identify and trace the flow of information within the collection. This also helps to forecast emerging research topics in the area of the study. Classification based on similarity information extracted from author information leads to identification of the different author collaboration groups within the collection. This is because researchers usually collaborate with others within similar area of expertise. In a similar manner, classification based on similarity information obtained from word content analysis can be used to classify articles according to their content similarity. However, this needs extra human effort and expert knowledge to resolve ambiguities introduced by the high diversity and noise within the natural language.

This research proposes a new method of classification of scientific literatures using a fused similarity matrix obtained from multiple sources of similarity matrices discussed above. A genetic algorithm based search method was used to search for the similarity information fusion parameters. Genetic algorithm was chosen as a search method in order to make the proposed method scalable to cases in which there are many similarity information sources with minimal computational complexity. Minimization of the coefficient of the Pareto Distribution for index terms within the final clusters formed was used as a fitness function for the genetic search. The final parameters returned by the genetic search algorithm were later used to fuse similarity matrices obtained from citation, author, and word content analysis. This fused matrix was passed to an agglomerative hierarchical clustering routine and a hierarchical time line visualization method was used to show the results. A simulated annealing based optimization is performed on the hierarchical time line visualization for a better understanding of the result. The results obtained using this method show that incorporation of similarity information from multiple sources helps to achieve a better classification that can be used to understand and further explore a collection of scientific literatures in an effective way.

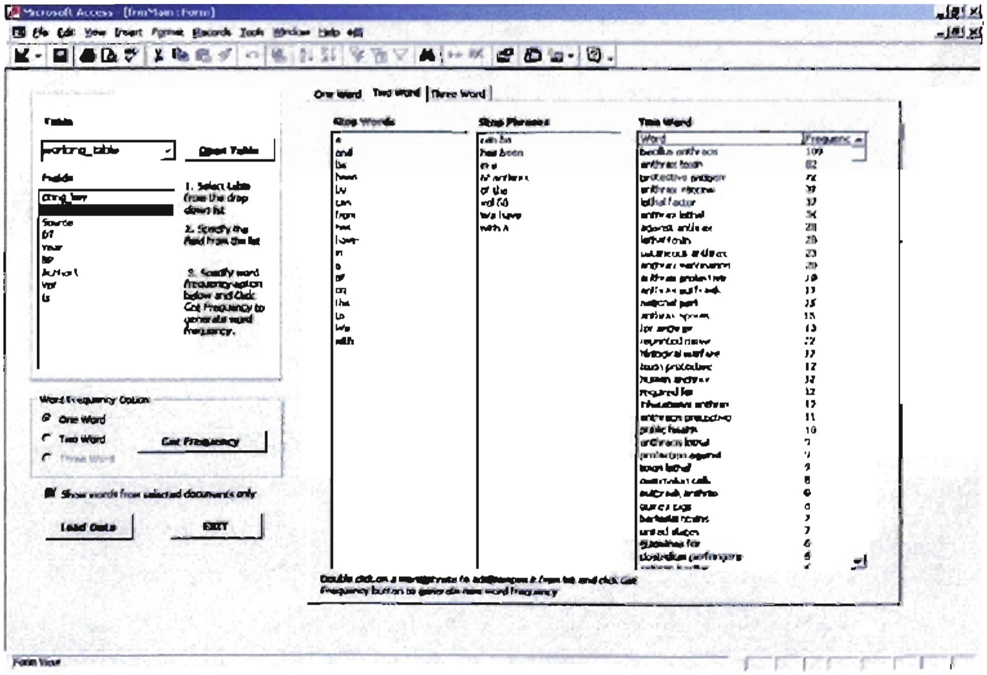
6.2 Suggested Future Work

Some of the recommendations for future work on this research include the following ideas. One major advance in this research involves applying the proposed method of utilizing multiple similarity information based classification to a collection of free text documents including sources from newspaper articles, web pages, and financial transactions. This is mainly dependent on the accuracy of the similarity information matrix computation. This would enable one to derive a complete view and understanding of a subject matter of interest based on knowledge extracted from all available sources. For example, in the case study presented earlier in Chapter 5, the knowledge acquired on the subject of "anthrax anthracis" was limited only to the content of the articles that we obtained from the ISI Science Citation Index library. However, if other sources, such as news releases and non-scientific peoples' opinion and experience, were added to the collection, the result might be able to give the researcher a more sophisticated, real-life understanding of the subject under study. This can also make the system to be used in intelligence applications where information from different sources is required in an organized manner to facilitate link discovery. Another future area of development is automatic generation of cluster labels, which is currently done manually. This needs taking careful consideration and sound judgment on the content of titles, abstracts, word frequency analysis and citation patterns.

APPENDIX A

A.1 Microsoft Access Database Application

This research involved storing of text data into a database. A Microsoft Access database application was developed for this purpose. The application was used to parse and input tagged text source data into its tables. It also had the functionality of generating one, two and three word frequency summary results excluding a user specified list of stop words for documents of interest specified by the user. Several SQL queries were also included in this application to assist in data retrieval and presentation in the MATLAB program. The main user interface of this database application is shown in Figure A.1.



1. Select table from the drop down list.

2. Specify the field from the list

3. Specify word frequency option below and Click Get Frequency to generate word frequency.

● Show words from selected documents only

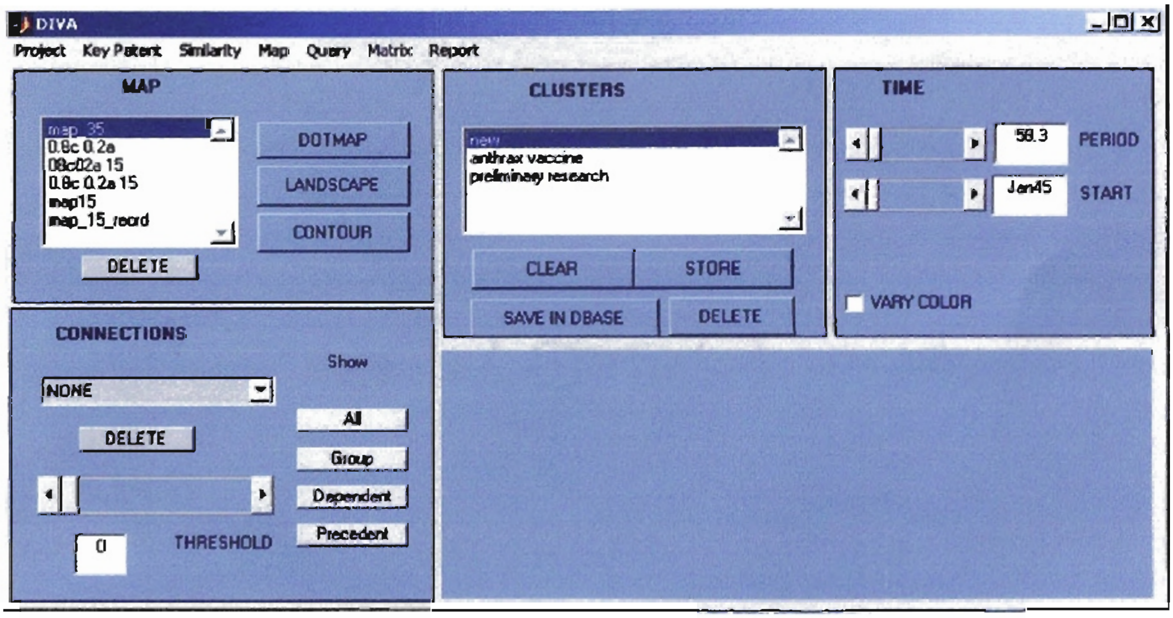
Double click on a word/phrase to add/remove it from list and click Get Frequency button to generate new word frequency

Word	Frequency
bacillus anthracis	109
anthrax toxin	82
protective antigen	72
anthrax vaccine	39
lethal factor	37
anthrax lethal	34
against anthrax	28
lethal toxin	28
cutaneous anthrax	23
[illegible]	20
anthrax protective	19
[illegible]	17
[illegible]	15
anthrax spores	15
for anthrax	13
[illegible]	12
biological warfare	12
toxin production	12
human anthrax	12
required for	12
inhalational anthrax	12
anthrax producing	11
public health	10
anthracis lethal	7
protection against	9
toxin lethal	9
mammalian cells	8
[illegible]	6
[illegible]	6
bacterial toxins	7
united states	7
guidelines for	6
clostridium perfringens	5

Figure A.1 Microsoft Access database application user interface

A.2 User Interface of MATLAB Program

The MATLAB program that served as a main tool for this research is called DIVA, Database Information Visualization and Analysis software. It is equipped with several functionalities that would allow users to access data through an Open Database Connectivity (ODBC) in order to analyze and visualize it. The main user interface of DIVA is shown in Figure A.2 below.



map_35
0.8c 0.2a
08c02a 15
0.8c 0.2a 15
map15
map_15_recrd

new
anthrax vaccine
preliminary research

PERIOD: 58.3
START: Jan45
□ VARY COLOR

NONE

THRESHOLD: 0

Figure A.2 The DIVA user interface

This user interface allows saving, retrieving and managing of files generated by DIVA. The *map* section lists stored two-dimensional maps in the current project that can be displayed at any time. The *connection* section lists different types of stored connection similarity matrices that can be used to classify and visualize the data. The *clusters* section can be used to store groups of interest into variables for later use. The *time* section is used to perform analysis based on temporal information. A typical map example of DIVA output is shown in Figure A.3 below.

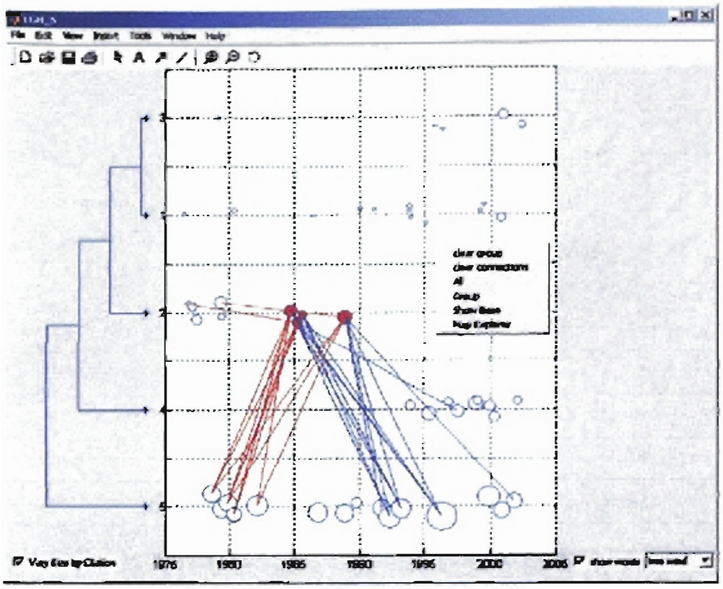


Figure A.3 Two-dimensional map display of DIVA output

Each dot on the map corresponds to a particular document and the dots highlighted in red indicate documents being selected. Lines represent similarity, if shown in green, or citation link, shown in red for backward citation and blue for forward citation. Right clicking on the map lists some available functions. The option "vary size by citation" on the left bottom corner allows the user to vary the size of the dots based on number of citations. The option, "show words" on the right bottom corner allows the user to display the most frequent words within documents upon selection by drawing a rectangle around them Clicking the '+' sign at the end of the each legs of the hierarchical tree structure performs further classification on the contents of the corresponding clusters and plots a new zoomed-in version of the map. For more details on DIVA, please refer to [19].

BIBLIOGRAPHY

[1] A. Griffiths, L. A. Robinson, and P. Willett, "Hierarchic agglomerative clustering methods for automatic document classification," *Journal of Documentation*, vol. 40, pp. 175-205, 1984.

[2] S. A. Morris, G. G. Yen, Z. Wu, and B. Asnake, "Timeline visualization of research fronts," *Journal of the American Society for Information Science and Technology*, vol. 54, pp. 413-422, 2003.

[3] Y. Zhao and G. Karypis, "Criterion functions for document clustering: Experiments and analysis," *University of Minnesota, Department of Computer Science*, Minneapolis, MN 01-40, 2002.

[4] G. Salton, *Automatic text processing: The transformation, analysis, and retrieval of information by computer*: Addison-Wesley Publishing Company, 1989.

[5] N. Fuhr, "Probabilistic models in information retrieval," *Computer Journal*, vol. 35, pp. 243-255, 1992.

[6] G. Salton, *The SMART retrieval system*: Prentice-Hall, Inc., 1971.

[7] D. Hiemstra and A. P. de Vries, "Relating the new language models of information retrieval to the traditional retrieval models," *University of Twente, CTIT* TR-CTIT-00-09, 2000.

[8] J. H. Lee, W. Y. Kim, M. H. Kim, and Y. J. Lee, "On the evaluation of Boolean operators in the extended Boolean retrieval framework," *presented at Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pittsburgh, PA, 1993.

[9] O.-W. Kwon, M.-C. Kim, and K.-S. Choi, "Query expansion using domain-adapted, weighted thesaurus in an extended Boolean model CIKM 94," *presented at Proceedings of the Third International Conference on Information and Knowledge Management*, Gaithersburg, MD, 1994.

[10] S. T. Dumais, T. K. Landauer, and M. L. Littman, "Automatic cross-linguistic information retrieval using latent semantic indexing," *presented at In proceedings of the ACM SIGIR '96 Workshop on Cross-Linguistic Information Retrieval*, Zurich, Switzerland, 1996.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391-407, 1990.

[12] T. K. Landauer and M. L. Littman, "Fully automatic cross-language document retrieval using latent semantic indexing," *presented at In Proceedings of the Sixth Annual conference of the University of Waterloo Centre for the New Oxford English Dictionary and Text Research*, 1990.

[13] K. Cios, W. Pedrycz, and R. Swiniarski, *Data mining: Methods for knowledge discovery*: Kluwer Academic Publishers, 1988.

[14] Y. H. Li and A. K. Jain, "Classification of text documents," *The Computer Journal*, vol. 41, pp. 537-546, 1998.

[15] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," *presented at International Joint Conference on Neural Networks*, Anchorage, Alaska, 1998.

[16] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, pp. 130-137, 1980.

[17] A. D. Gordon, *Classification*, 2nd ed: Campman & Hall/CRC, 1998.

[18] S. Lawrence, C. L. Giles, and K. Bollacker, "Digital libraries and autonomous citation indexing," *IEEE Computer*, vol. 32, pp. 67-71, 1999.

[19] S. Morris, C. DeYoung, Z. Wu, S. Salman, and D. Yemenu, "DIVA: A visualization system for exploring document databases for technology forecasting," *Computers and Industrial Engineering*, vol. 43, pp. 841-862, 2002.

[20] H. Small, "Update on science mapping: Creating large document spaces," *Scientometrics*, vol. 38, pp. 275-293, 1997.

[21] H. D. White and B. Griffiths, C., "Authors as markers of intellectual space: Co-citation in studies of science, technology and society," *Journal of Documentation*, vol. 38, pp. 255-272, 1982.

[22] H. D. White and K. W. McCain, "Visualizing a discipline: An author co-citation analysis of information science, 1972-1995," *Journal of the American Society for Information Science*, vol. 49, pp. 327-355, 1998.

[23] C. Chen, "Visualizing semantic spaces and author co-citation networks in digital libraries," *Information Processing and Management*, vol. 35, pp. 401-420, 1999.

[24] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of massive document collection," *IEEE Transactions on Neural Networks*, vol. 11, pp. 574-585, 2000.

[25] K. Lagus, "Text mining with the WEBSOM," in *Acta Polytechnica Scandinavica*, vol. 110, *Mathematics and Computing Series*: Finnish Academies of Technology, 2000.

[26] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 264-322, 1999.

[27] Mathworks, *MATLAB: Statistics toolbox user's guide. Version 3*: The Mathworks Inc., 2000.

[28] J. S. Rojer Jang and N. Gulley, *MATLAB: Fuzzy logic toolbox user's guide. Version 1*: The Mathworks Inc., 1995.

[29] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural network design*: PWS Publishing Company, 1996.

[30] T. Kohonen, *Self-organization and associative memory. Third edition*: Springer-Verlag, 1989.

[31] S. A. Morris, Z. Wu, and G. G. Yen, "A SOM mapping technique for visualizing documents in a database," *presented at International Joint Conference on Neural Networks*, Washington D.C., 2001.

[32] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, pp. 586-600, 2000.

[33] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*: Addison-Wesley Publishing Company, Enc., 1989.

[34] J. H. Holland, K. F. Holyoak, R. E. Nisbett, and P. R. Thangard, *Induction: Processes of inference, learning, and discovery*. London, England: The MIT Press, 1986.

[35] N. L. Johnson, S. Kotz, and A. W. Kemp, *Univariate discrete distributions*: John Wiley & Sons, Inc., 1992.

[36] S. Kirkpatrick, C. Gelatt, and V. M., "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.

[31] S. A. Morris, Z. Wu, and G. G. Yen, "A SOM mapping technique for visualizing documents in a database," *presented at International Joint Conference on Neural Networks*, Washington D.C., 2001.

[32] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, pp. 586-600, 2000.

[33] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*: Addison-Wesley Publishing Company, Enc., 1989.

[34] J. H. Holland, K. F. Holyoak, R. E. Nisbett, and P. R. Thangard, *Induction: Processes of inference, learning, and discovery*. London, England: The MIT Press, 1986.

[35] N. L. Johnson, S. Kotz, and A. W. Kemp, *Univariate discrete distributions*: John Wiley & Sons, Inc., 1992.

[36] S. Kirkpatrick, C. Gelatt, and V. M., "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.

VITA

Benyam Tesfaye Asnake

Candidate of the Degree of

Masters of Science

Title of Study: AUTOMATIC SCIENTIFIC LITERATURE CLASSIFICATION USING MULTIPLE INFORMATION SOURCES FOR DATA MINING PURPOSES

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Addis Ababa, Ethiopia, 1978, the son of Tesfaye Asnake and Yewelsew Abebe.

Education: Graduated from Comboni Senior Secondary High School, Awasa, Ethiopia, in May 1995; received Bachelor of Science degree in Electrical and Computer Engineering from Addis Ababa University, Addis Ababa, Ethiopia in June 2000. Completed the requirements for the Master of Science degree with a major in Electrical Engineering at Oklahoma State University in May, 2003.

Experience: Employed as a Commodity Tracking System Controller, July 2000 to July 2001 by the United Nations, World Food Programme, Ethiopia; employed by Oklahoma State University, School of Electrical and Computer Engineering as a research and teaching assistant, August 2001 to May 2003.

Professional Membership: Institute of Electrical and Electronic Engineers, Phi Kappa Phi, national honorary society.

#7

Name: Benyam Asnake Date of Degree: May, 2003

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: AUTOMATIC SCIENTIFIC LITERATURE CLASSIFICATION USING MULTIPLE INFORMATION SOURCES FOR DATA-MINING PURPOSES

Pages in Study: 69 Candidate for the Degree of Master of Science

Major Field: Electrical Engineering

Scope and Method of Study: This research concentrated on automatic classification of scientific literatures retrieved in a particular field of study. The classification methodology proposed in this work aimed at utilizing different similarity information matrices extracted from citation, author, and term frequency analysis. These similarity matrices were fused into one generalized similarity matrix by using parameters obtained using a genetic search algorithm. The final similarity information matrix was passed to an agglomerative hierarchical clustering routine to classify the articles. A simulated annealing based optimization was performed on the output of the hierarchical classification for optimized visualization and interpretation of the result.

Findings and Conclusions: This work was able to demonstrate that multiple similarity information could be used to classify scientific literatures for a better understanding. The similarity information fusion technique introduced was able to achieve an optimal classification of the collection. A simulated annealing based optimization technique for visualization was also introduced which helped achieve an optimal visualization. The work showed that the proposed method was able to identify the main research areas, emerging fields, major authors and their area of excellence within a collection of scientific literatures distinctly and in an efficient manner.

ADVISER'S APPROVAL: ____________________