A CONTEXT VECTOR-BASED SELF ORGANIZING MAP FOR INFORMATION VISUALIZATION

David A. Rushall, Marc R. Ilgen
HNC Software, Inc., 5930 Cornerstone Court West, San Diego, CA 92121 USA
email: dar@hnc.com, mri@hnc.com

ABSTRACT

HNC Software, Inc. has developed a system called DOCUVERSE for visualizing the information content of large textual corpora. The system is built around two separate neural network methodologies: context vectors and self organizing maps. Context vectors (CVs) are high dimensional information representations that encode the semantic content of the textual entities they represent. Self organizing maps (SOMs) are capable of transforming an input, high dimensional signal space into a much lower (usually two or three) dimensional output space useful for visualization. Related information themes contained in the corpus, depicted graphically, are presented in spatial proximity to one another. Neither process requires human intervention, nor an external knowledge base. Together, these neural network techniques can be utilized to automatically identify the relevant information themes present in a corpus, and present those themes to the user in a intuitive visual form.

1. INTRODUCTION

In recent years there has been an explosion in the amount of information available on-line. Much of this explosion has been fueled by the spectacular growth of the Internet and especially the World Wide Web. Along with this spectacular growth has come new challenges for effectively locating on-line information, especially when browsing rather than performing a directed search for a specific piece of information. Key word and linguistically based directed search engines offer some ability to present relevant information to the user, and have resulted in useful Internet products such as Yahoo [1], Lycos [2], and Alta Vista[3]. However, these engines suffer from the fact that they require the user to specify a query of limited length and they offer no visual interface for browsing. To solve the problem of browsing the information space in order to find information of interest, new techniques for data retrieval and presentation must be developed.

HNC has developed an underlying information representation technology and a concept for information visualization that can solve the problem of effectively browsing large textual corpora. As part of HNC's involvement in the ARPA sponsored TIPSTER program, HNC has developed a neural network technique that can learn word level relationships from free text. This capability is based upon an approach called context vectors which encodes the meaning and context of words and documents in the form of unit vectors in a high dimensional vector space. Furthermore, as part of HNC's involvement in the US intelligence community-sponsored P1000 visualization effort, HNC has applied a secondary neural network process, the Self Organizing Map (SOM) [4], which uses the document context vectors to build a visual representation of the information content of the corpus. The combination of these technologies allows users to effectively browse the information space, to locate related documents, and to discover relationships between different themes in the information space.

The remainder of this paper is organized as follows. Section 2 presents an overview of both context vectors and the Self Organizing Map. Section 3 presents the DOCUVERSE system and presents the user interface, automatic region finding and region labeling, information retrieval and document highlighting, and temporal analysis of the information space. Finally, Section 4 presents some concluding remarks and directions for future research.

2. TECHNICAL BACKGROUND

The DOCUVERSE system is based on two technologies: context vectors and the SOM. Context vector technology was developed at HNC and has been demonstrated to be highly effective for such tasks as text retrieval (using a system called MatchPlus) [5],

text routing [6], and image retrieval [7]. SOM technology was originally developed by T. Kohonen and has been used throughout the neural network community as a method for representing information in a manner suitable for visualization [8]. The following subsections present an overview of each of these technologies.

2.1 Context Vectors

The key technical feature of context vector technology is the representation of terms, documents, and queries by high dimensional vectors consisting of real-valued numbers or components. These vectors are constrained to be unit vectors in the high dimensional vector space. Since both terms (words or stems) and documents are represented in the same frame of reference, this allows several unique operations. All operations in MatchPlus are based on geometry of these high dimensional spaces [9]. Specifically, closeness in the space is equivalent to closeness in subject content. A neural network-based learning algorithm is designed to adjust word vectors such that *terms that are used in a similar context will have vectors that point in similar directions.* The determination of similar context is based upon the use of word stem co-occurrence statistics. Once trained, these word stem context vectors are used as building blocks for creating document and query context vectors. Specifically, context vectors for documents and queries are formed as the normalized weighted sum of the word stem context vectors for the word stems found in the document or query. This process results in the fact that context vectors for documents with similar subject content will point in similar directions. This vector representation of information content can thus be used for document retrieval, routing, document clustering (self organizing subject index) and other text processing.

2.2 Kohonen's Self Organizing Map

The concept of self organizing maps was first developed by Tuevo Kohonen in 1981 at the University of Helsinki. Kohonen demonstrated that a system could be taught to organize the data it was given, without the need for supervision or external intervention, through the use of competitive learning. Typically, the SOM consists of a collection of nodes arranged in a regular two dimensional grid. Each node corresponds to a cluster centroid vector for the high dimensional input vector space. A self-organizing training process is used to adjust the node vector components in an iterative fashion. Upon completion of training, the SOM node vectors have the property that node vectors that are close in the high dimensional vector space will be close in the two dimensional grid space, or "map space." This training process is described in more detail below.

2.2.1 Training the SOM

Assume the input space is comprised of N vectors, each of dimension n. Furthermore, assume a two dimensional array of "nodes" that will be trained to represent the N input vectors. Each of these nodes is also a vector of dimension n. Figure 1 depicts this arbitrary array of nodes, in this case, a 5-by-5 regularly spaced array of 25 nodes. Each node is uniquely identified by it's (i,j) position.

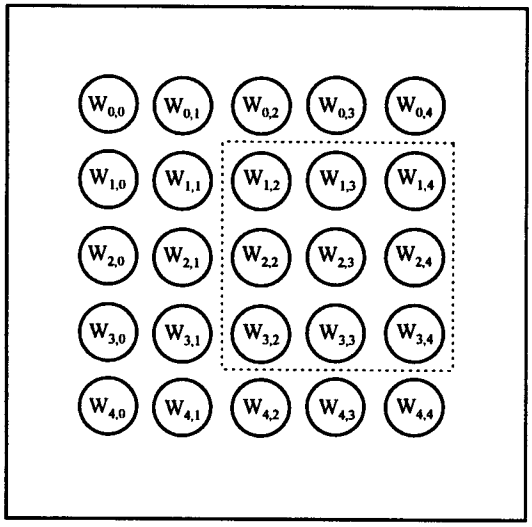

Figure 1. An array of map nodes for self organization.

Before training, the node vectors are assigned random values. That is, a pseudo random number generator is used to assign each node in the map a random unit vector of dimension n. In the case where n is very large (~300), these initial conditions represent a quasi-orthogonal state, i.e., each unit vector is approximately orthogonal to each other unit vector.

The SOM training algorithm is based on a simple, iterative comparison process, where the N input vectors are compared to each of the node vectors in the map. In essence, the node vectors are competing to have their values adjusted. The idea is to find the node vector that is "nearest" (in vector space) to the input vector. The node vector that is nearest is deemed the "winner" of this competition,

and is rewarded by having its vector adjusted. The adjustment comes in the form of moving the winning node vector in the direction of the input vector. The SOM extends this simple competitive learning process to include updates for the "neighbor" nodes to the winning node. These neighbor nodes are nodes that are close (in the map space) to the winning node. For example, in Figure 1, the neighborhood of node $W_{2,3}$ is depicted, and is defined as those nodes that are within one row or one column of node $W_{2,3}$. Neighborhoods can be larger or smaller; this is just one example. The updates for these neighbor nodes are smaller than the updates for the winning node, and the size of the neighbor node update is smaller for neighbors that are farther away (in map space) from the winning node. The inclusion of neighbor updates results in the organization of the information into a form suitable for visualization. The algorithm can be represented in pseudo-code as:

```
For each input vector V
  Find node vector W_ij that is closest to V
    ( | W_ij - V | < | W_k,l - V |  ∀k, l)
  Update W_ij according to W_ij = W_ij + α( V - W_ij )
  Find nodes that are close to node i,j in map space
  For each of these "neighbor" nodes W_k,l
    Update W_k,l according to W_k,l = W_k,l + sα( V - W_k,l )
    where ( 0.0 ≤ s ≤ 1.0 )
  End loop over neighbor nodes
End loop over input vectors
```

The size of the adjustment, α, will determine how quickly the map space node vectors will converge to an accurate representation of the input space vectors. One loop through the input vector set is not sufficient to train the node vectors. It is necessary to perform this loop hundreds, and possibly thousands, of times before training is completed. The value of the parameter s for updating neighbor nodes is determined by a Gaussian function based on the nearness of the neighbor node to the winning node. Therefore, close neighbors will be updated, or adjusted, more than neighbors that are further away.

2.2.2 Win Frequency and Conscience

An ideal characteristic of this training is to have the map node vectors win the competition with equal probabilities. Unfortunately, this is not the case for the standard SOM algorithm. A consequence of this type of training is that some map nodes may never win the competition. This will result in a less useful representation of the input space. To eliminate this undesirable effect, DeSieno [10] has developed an improved competitive learning algorithm that makes use of the idea of "conscience". The conscience mechanism allows nodes that are observed to rarely win the competition to subsequently win more often, and it prevents nodes that frequently win the competition from subsequently winning too often.

The conscience mechanism is employed as a second competition based on the outcome of the first competition described above in the self organizing algorithm. Before conducting the second competition, the conscience mechanism creates a bias factor for each node. The value of the bias is determined by the running statistics kept on the first competition. Nodes that normally lose the first competition are given favorable biases, and those that normally win are given unfavorable biases. The winner of this second competition is determined upon the basis of biased distance to the input vector. This biasing enforces an equiprobable winning distribution and results in a more useful clustering of the input information space.

2.2.3 Computation Issues

Earlier we alluded to the fact that these algorithms are computationally intensive. This intensity depends upon three factors. One is the dimensionality of the vectors. Using higher dimensioned vectors (~1000) is possible, but adds to the timely computation problem. Likewise, the number of input vectors, as well as the number of map node vectors, will determine the scale of the problem. Fortunately, the nature of these algorithms is well suited for parallel processing architectures. Therefore, scalability of the algorithm depends on the number of processors that can be used to compute a solution.

HNC has developed a hardware architecture that is designed to handle neural networks, and in particular, the compute intensive processes they model. The hardware is a SIMD numerical array processor (SNAP), in essence a floating-point parallel array processor. The SNAP is ideally suited to determine the computationally intensive solution required by the SOM algorithms.

The SNAP comes in a variety of configurations. The fastest SNAP available, the SNAP-64, has 64 processors, and delivers an unmatched price-to-performance ratio of around $20 per megaflop. At its peak, the SNAP computes at a rate of 2.56 gigaflops. This type of performance enables HNC to develop and deliver the compute intensive solutions to a wide variety of problems, including information visualization.

3. DOCUVERSE

The DOCUVERSE system, developed as part of the IC P1000 research effort at HNC, is an ongoing research and development effort with a goal of providing users with a tool to quickly and easily assess the information content of large textual corpora. DOCUVERSE is based on the context vector technology foundation developed at HNC over the past few years, and additionally provides a visual interface that allows the user to browse the information space in a visually appealing fashion.

Figure 2 presents the process by which a text corpus is transformed into some intuitive visual paradigm that users can easily relate to and understand. The initial neural network process is shown along the top part of Figure 2. The process flow indicates that some set of textual data, the training text, is used to obtain context vectors for the vocabulary set contained in the training text. This process involves a preliminary step of word "stemming" and stop list removal. Stemming is the process of representing similar word forms as the base form of the words (i.e. words like driver, driving, drives, driven, and drove are all stemmed to the word drive). Stop list removal refers to the removal of words with high frequency occurrence and little meaning in the training text (i.e. words like the, of, and, etc.). After preprocessing, context vectors for the remaining word stems are learned and stored into a database.

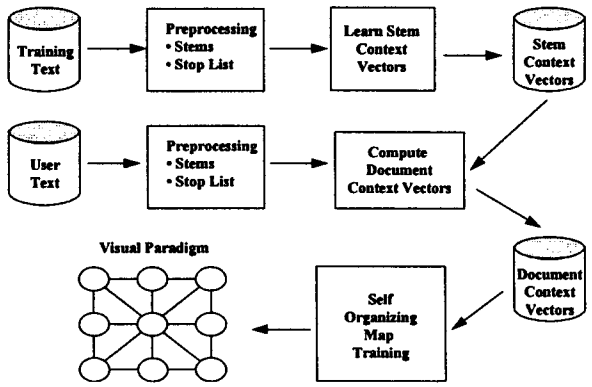

Figure 2. The process of transforming textual data into an intuitive, graphical visual.

At this point, the system is ready to process the user's desired corpus: Note that the corpus to be visualized does not need to be the same corpus that the system was trained with. It is helpful, however, if the training corpus is statistically representative of the corpora that will be visualized.

As shown in Figure 2, the control flow now switches to the middle part of the diagram, where the user identifies the corpus to be visualized. This text is preprocessed in the same exact manner as the training text was preprocessed. After this step, one pass through each document is all that is required to calculate a context vector for each document. The stem vectors learned from the training text are used to compute the context vectors for the user's text. These are stored in a document context vector database. It is these context vectors that are given to the self organizing map for visualization.

2.3 The Interface

The DOCUVERSE interface presents the user with an array of nodes not unlike the array of nodes depicted in Figure 1. The size of the array is configurable by the user, but the default is a 20-by-20 array of nodes. Recall that each of these 400 nodes has a context vector associated with it, and that the context vectors have been adjusted to represent the prevalent themes in the corpus. Therefore each node represents an information theme contained in the corpus. It is important to note that it is not necessary that each node have a different theme. Nodes can have similar themes, and in fact, the same theme if there is a relatively large amount of information pertaining to that particular theme within the corpus.

This discussion raises the question as to how the nodes reflect the amount of information present for the theme they represent. After experimenting with various paradigms such as color or icons, we've concluded that the size, or radius, of the nodes best conveys this information. Large nodes imply a relatively large number of documents for the given information theme. Small nodes imply a relatively small number of documents for the given information theme.

The way in which the system measures not only the amount of information for a theme, but also the similarity of themes, documents, words, or any free text, is through the use of the vector dot product operation. Recall that each of the aforementioned textual entities is associated with a context vector. Similar entities have context vectors that point in similar directions. The dot product for similar direction vectors will be close to 1.0, while dissimilar vectors will have dot products that are near zero.

Figure 3 depicts what we call the "corpus integral". This is the broad view of information

content for the entire corpus. It is an attempt to graphically illustrate, through various sized information nodes, the entire set of prevalent themes contained in the corpus. Again, nodes that are large represent the themes that occur with the highest frequency and volume in the corpus. Nodes that are small, or not even visible (such as those in the upper left corner of the map), represent themes that occur with a much lower frequency and volume, relatively speaking.

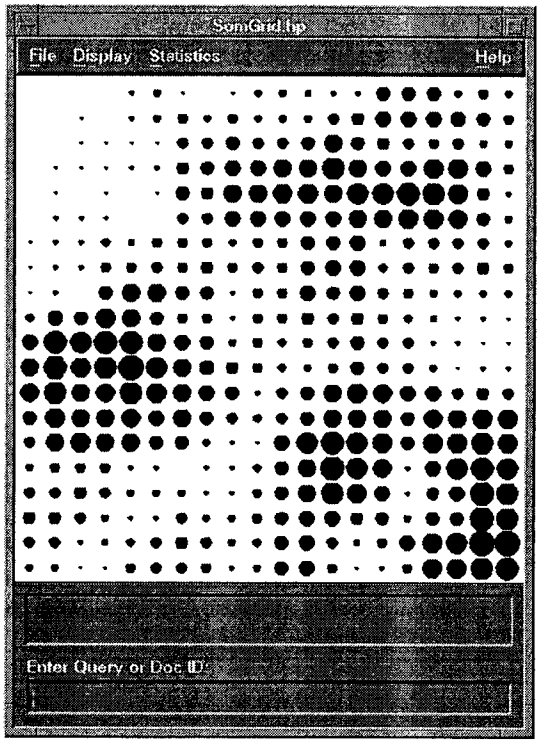

Figure 3. The corpus integral.

The corpus integral is computed by summing the dot products of each document context vector with each node context vector. The summed dot products for each node are used to determine the size of the node. In pseudo-code,

```
For each node vector W_ij
  Node_size_ij = 0
  For each document vector V
    Node_size_ij += W_ij • V
  End loop over document vectors
  Node_size_ij /= number of document vectors
End loop over node vectors
```

The corpus integral is very useful in that the user knows, at a glance, which themes are present in the corpus. By "mousing" on a node (i.e. clicking the mouse button once on a node), a pop-up menu reveals, among other choices, the information theme the node represents.

The corpus that was used to generate the integral depicted in Figure 3 is a set of over 17,700 documents. The documents are news reports taken directly off the AP News Wire during a four month span in 1990.

Other system capabilities, discussed in the sections below, allow the user to do a variety of information assimilation and information gathering tasks. For example, an undirected information search, commonly referred to as browsing, is made even easier when using the automatic region finding and labeling mechanism. Searching for specific information is also supported so that the user can request, in free text form, any desired information. A tool for visualizing information in the time domain is also provided.

2.4 Automatic Region Finding and Labeling

As we stated earlier, nodes that are near one another on the map have the property that they represent similar themes of information. We can exploit this fact to have the system automatically group the nodes into regions of similar themes. This can be thought of as a clustering of the clusters. Furthermore, the system will automatically generate an appropriate name, or label for the region. Consider Figures 4a and 4b.

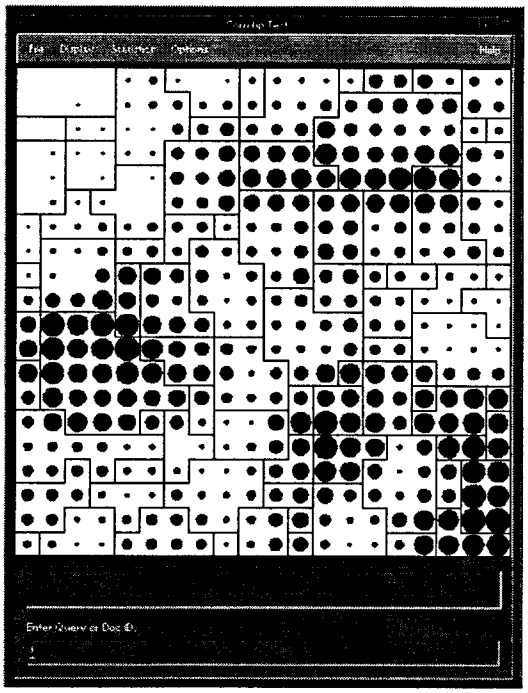

Figure 4a. The automatically generated regions

Figures 4a and 4b show one of the many ways that a user can make use of the automatic region generation. They depict the option of selecting all regions found on the map. Figure 4a shows all the regions that were found by the system. The regions are outlines drawn around sets of nodes. Figure 4b shows a second dialog window that is used to display the labels for the regions. Although all the labels are not visible in the dialog, the region algorithm found a total of 84 distinct regions for the self organized map of the 17,700 AP News Wire documents.

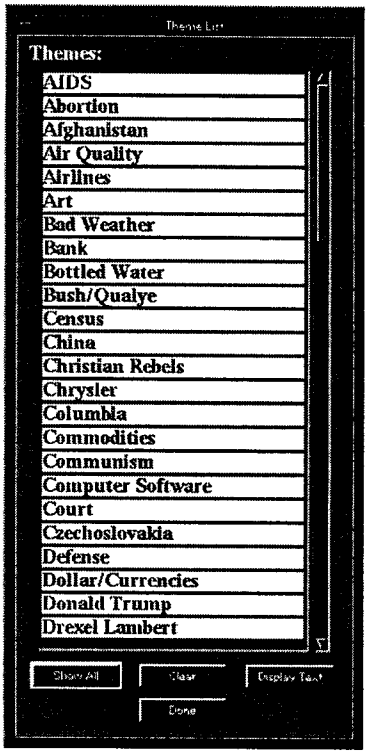

Figure 4b. The automatically generated labels.

Instead of showing all regions at once, a user can select regions of interest from either the map or the label dialog. If a user is interested in a particular area on the map, the region outlines for the nodes of interest can be toggled on or off by the pop-up menu provided on each node.

Alternatively, the user can peruse the list of region labels and select them directly from the list. This will draw the region outline on the map.

The algorithm for region finding and labeling is a two step process. The first step involves finding the regions. Initially, each node is given its own region. An iterative algorithm compares a region to every other region on the map. The comparison is a vector dot product operation. If the regions are similar enough (i.e., if the dot product between the two regions exceeds some threshold), the two regions are merged into one region. This process is repeated until no region combining occurs.

The next step involves finding an appropriate label for each region. First, a context vector is computed for the region. This is done by taking the weighted average of the context vectors belonging to the nodes in the region. This centroid region context

vector is compared to all of the stem word context vectors in the vocabulary. The stem word vector that results in the highest dot product with the centroid region vector becomes the label for the region. Because of the stemming process, some of the stem words are truncated. Therefore, the user is given the ability to edit the labels to put them in correct grammatical form.

2.5 Information Retrieval and Document Highlighting

DOCUVERSE makes use of a rich set of information retrieval functionality. This functionality was inherited from another system developed at HNC called MatchPlus [5]. MatchPlus focuses on information retrieval from large textual corpora.

When a user desires a more focused search for specific information, it is easily accomplished in a variety of ways. If a user has identified a map node representing an interesting theme of information, the user can select, from the node pop-up menu, an option to retrieve documents pertaining to the theme. The system uses the node's context vector to perform dot products with every document context vector in the corpus. The user is presented with a ranked list of the most relevant documents. The list is presented in a window with the document ID, the value of the dot product, and the first line of text in the document. Figure 5 shows an example of the ranked list.

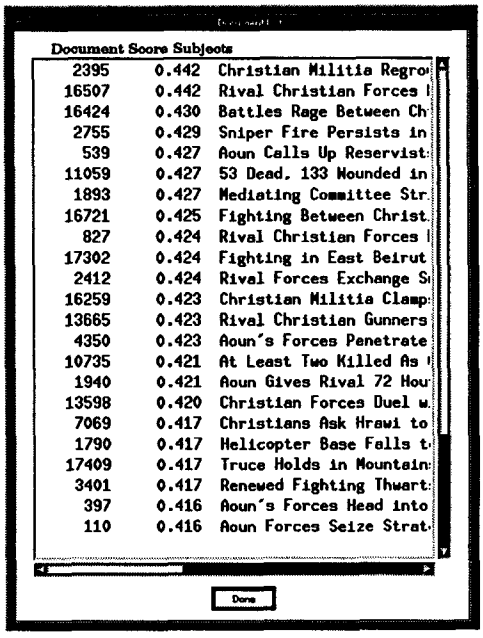

Figure 5. A ranked list of documents.

Alternatively, rather than using a node for a query, the user can type free text into a window and submit the free text as a query. The system converts the free text into a context vector, and the same retrieval process is performed. Yet another possibility is for the user to use an entire document as a query. Regardless of the method, the retrieval is done via context vector comparisons.

Once a ranked list of documents has been retrieved, the user can select any document from the list to view. The document selected will appear in a separate window, along with a highlighting tool to further examine the relevant parts of the document.

The highlight tool segments the document into 5-line paragraphs. The user is presented with another window containing a histogram, where each interval of the histogram corresponds to each of the paragraphs in the document, and the height of the interval corresponds to the dot product of the paragraph with the query that was issued. This tool provides the user with a tool to quickly and easily locate the most relevant portions of any document.

2.6 Temporal Analysis

With corpora comprised of periodically released information, it might be useful to visualize the information in the time domain. The DOCUVERSE temporal analysis tool makes this possible. Documents are segmented into user-defined time intervals, typically one hour, one day, or one week, depending upon the nature of the data. By performing a cumulative dot product operation for each document in the interval with the corpus integral, we can obtain a visual summary of the information content of the documents received during that time interval. The resulting time series of information themes can be viewed in rapid succession. The user is provided a media-player type interface with buttons for "play', "stop", and stepping forward and backward. Using the step buttons, the user can manually step through each time increment, or alternatively, the play button will rapidly step through the time increments in succession.

When using the temporal tool on the 17,700 AP News Wire documents, weekly cycles are easily identified. By stepping through the data in one day increments, the weekdays (Monday through Friday) are identified by three predominate regions pertaining to themes like banks, stocks, world news, and taxation. The next two days, the weekend, show maps with two predominate regions, pertaining to themes like music, TV, movies, and various other entertainment themes.

4. SUMMARY

As we continue through the information age, tools such as DOCUVERSE will no longer be considered luxuries, but rather necessities. HNC has developed DOCUVERSE as merely a proof of concept system. We feel that this technology is far from realizing its full potential. As information visualization technology evolves and matures, so too will tools like DOCUVERSE.

HNC is in the process of exploring new ways to visualize this powerful information representation technology. One area of interest is developing three dimensional SOMs. The user will be presented with a spherical array of nodes, representing the "world" of information. Used in conjunction with flat maps, the user would have a hierarchical SOM capable visualizing information at various levels of resolution.

It is clear that in terms of visualization, "one size fits all" does not apply. What is intuitively obvious to one user is unclear and convoluted to another. Realizing this, we have identified numerous browsing paradigms to appeal to a broader audience. Tools like the Virtual Reality Modeling Language (VRML) are well suited for use with this technology for visualizing information on the World Wide Web, and will aid us as we strive to improve information visualization.

REFERENCES

[1] Yahoo!, http://www.yahoo.com/

[2] Lycos, http://www.lycos.com/

[3] Alta Vista, http://altavista.digital.com/

[4] Kohonen, T., Self-Organizing Maps, Springer-Verlag, Berlin, 1995.

[5] Gallant, S.I., W. R. Caid, et al, "Feedback and Mixing Experiments with MatchPlus", Proceedings TREC-2 Conference, D. Harman, Ed, Gaithersburg, MD. Aug. 1993.

[6] Sasseen, R. V., J. L. Carleton, W. R. Caid, "CONVECTIS: A Context Vector-Based On-Line Indexing System", in *Proceedings IEEE Dual-Use Conference*, 1995.

[7] Pu, K. Q., C. Z. Ren, "Image/Text Automatic Indexing and Retrieval System Using Context Vector Approach," SPIE Volume 2606, 1995.

[8] Kohonen, et. al., http://websom.hut.fi/websom/

[9] Watson, G.S., "Statistics on Spheres", John Wiley and Sons, 1983.

[10] DeSieno, D., "Adding a Conscience to Competitive Learning", in *Proceedings of the International Conference on Neural Networks*, I, IEEE Press, NY, 1988.