Richard Freeman, Hujun Yin and Nigel M. Allinson, "Self-Organising Maps for Tree View Based Hierarchical Document Clustering", in Proceedings of the IEEE IJCNN'02, Honolulu, Hawaii, 12-17 May, 2002. vol. 2, pp. 1906-1911.

Self-Organising Maps for Tree View Based Hierarchical Document Clustering

Richard Freeman, Hujun Yin, Nigel M. Allinson.
University of Manchester Institute of Science and Technology (UMIST),
Department of Electrical Engineering and Electronics,
PO Box 88, Manchester, M60 1QD,
United Kingdom
Email: research@rfreeman.net, {H.Yin, allinson}@umist.ac.uk

Abstract – In this paper we investigate the use of Self-Organising Maps (SOM) for document clustering. Previous methods using the SOM to cluster documents have used two-dimensional maps. This paper presents a hierarchical and growing method using a series of one-dimensional maps instead. Using this type of SOM is an efficient method for clustering documents and browsing them in a dynamically generated tree of topics. These topics are automatically discovered for each cluster, based on the set of document in a particular cluster. We demonstrate the efficiency of the method using different sets of real world web documents.

I. INTRODUCTION

The growth of documents available digitally on corporate Intranets is continuously increasing. This makes it more and more difficult for company employees to manually organise, sort, and retrieve documents located on their corporate Intranet. In this paper we address this issue by proposing a method that can autonomously organise documents, using a series of independently trained Self-Organising Maps (SOM) to automatically cluster unstructured web documents. Features, words and terms will also be used interchangeably.

In section II we provide a brief introduction to the document clustering. Section III gives a description and review of the previous work on SOM and it's variations applied to the document-clustering problem. In section IV we will introduce the proposed method for dealing with the document clustering as well as some experimental results. Finally in section V we conclude and describe the future directions of this work.

II. DOCUMENT CLUSTERING

Document clustering is an area that involves automatically grouping related documents together. This is done without any prior training or external knowledge and is purely based on inferring suitable classes for a given set of documents.

A typical document clustering pre-processing phase uses the Vector Space Model (VSM) [1]. There are two main phases that are parsing and indexing. Parsing turns the text documents into a succession of words. The words are filtered using a basic "stop list" of common English words. This is used to discard words with little information, such as "the", "because", "it" that do not significantly contribute to discriminate between documents. A plural stemming or suffix stemming algorithm is then applied to those words [2]. In the Indexing phase, each document is then represented in the VSM where the frequency of occurrence of each word (or terms) in each document is recorded in a Documents Vs Terms matrix. These values are then generally weighted using the Term Frequency multiplied by the Inverse Document Frequency as shown in equation (1). This allows the less frequent terms to be given more weighting than the more frequent terms. Following Shannon's information theory the less frequent the word, the more information value it possess: this permits a better coverage of the input documents.

$$W_{ij} = tf_{ij} \cdot \log \frac{N}{df_i} \quad (1)$$

W_{ij} – Weight of term t_j in document d_i
tf_{ij} – Frequency of term t_j in document d_i
N – Total number of documents in collection
df_j – Number of document containing term t_j

Once the set of document vectors has been created we can use hierarchical or non-hierarchical techniques from Cluster Analysis. Hierarchical clustering places documents into a hierarchical structure that is built dynamically. Non-hierarchical methods or flat partitioning divide documents into a set of flat clusters. Document clusters are usually created based on a pre-defined criterion or an error measure between the documents. The most common error measures used for evaluating the similarity of two documents are the Manhattan distance, the Euclidean or cosine correlation. These measures are then used with the chosen clustering method, such as hierarchical methods like Single-Link method, Complete-Link method or Group Average Link methods [3].

The SOM offers a "neural" alternative to clustering with an additional topological preserving property. We shall now introduce the general SOM algorithm and then proceed with some of the previous work on document clustering using the SOM.

III. RELATED WORK

1) Introduction to the SOM

The Self-Organising Map (SOM) is one of the most widely applied Artificial Neural Networks (ANN) first introduced by Kohonen [4]. It has successfully been used in a variety of

applications including, data visualisation, data clustering, pattern recognition and data mining.

The SOM's objective in document clustering is to group the documents, which appear similar, close to one another and place the very different ones distant from one another. These output results are generally represented on a map (in two dimensions) or on a line (one dimension). In the basic SOM algorithm there are two basic steps, which are the search for the Best Matching Unit (BMU) and updating the BMU with it's neighbours. The BMU is found by computing the Euclidean distance between the input data vector x (document) and the reference vector m_i (weight) as show in equation (2).

$$BMU = \arg\min_i \{ \| x - m_i \| \}, i = 0, 1, \ldots n \quad (2)$$

Where n is the number of neurons in the SOM's feature map.

Once we have found the BMU we update the BMU and it's neighbouring nodes using:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)], t = 1, 2, 3 \ldots \quad (3)$$

Where:
t – discrete time constant
h_{ci} – neighbourhood function

The neighbourhood function h_{ci} used in equation (3), is a time decreasing function which determines to which extent the neighbours of the BMU will be updated. The extent of the neighbourhood is the radius and learning rate contribution, which should both decrease monotonically with time to allow convergence. The radius is simply the maximum distance at which the nodes from the BMU are affected. A typical smooth Gaussian neighbourhood kernel is given bellow in equation (4).

$$h_{ci} = a(t) \cdot \exp\left(- \frac{\| r_c - r_i \|^2}{2s^2(t)} \right) \quad (4)$$

Where:
$a(t)$ – learning rate function
$s(t)$ – kernel width function
$\| r_c - r_i \|^2$ – distance of BMU unit to current unit i

There are various functions used as the learning rate $a(t)$ and the kernel width functions $s(t)$. For further details about the SOM please refer to [4] and [5].

There are four major types of SOM that have been used for document clustering, visualisation or analysis. The first was to use a large two-dimensional map as in the WEBSOM [4]. The others added dynamic growth of the map and hierarchy connection of independently spun maps. There is also a combination of both growing and hierarchical SOM.

2) WEBSOM

The WEBSOM project was aimed to show that the SOM could be used as an effective graphical document exploration device [4]. In this method the words were first segmented in the input documents. Then the words with low frequency of occurrence were discarded and stop words were removed. The remaining words were then used as feature elements for word category maps, which captured some of the contextual information [6] and [7]. WEBSOM uses two layers of SOMs, the word category map and the document map. The first layer is the word category map, which learns to represent words using their average context vectors. The purpose of this map is to look for features that are useful to represent documents. These are then weighted using equation (1) and further blurred using convolution of a symmetric gaussian kernel. In the second layer called document map all the documents are clustered using a blurred histograms as input vectors for each document. In later updates the first layer called word category maps had been replaced by randomly projected histograms [8]. Good results have been reported based on a huge set of documents (1 million Usenet newsgroups, 6 million patents). The major interest in WEBSOM was the graphical map representation which could be navigated and help visualise the document clusters.

3) Hierarchical SOM

The hierarchical SOM (HSOM) works like the SOM algorithm, but starts with a top most level with all documents available for training. Once the top level is finished, it proceeds to further lower levels. However only documents that are associated with a parent node in the top map are used for each map in the lower level. The same process is then repeated for subsequent levels.

Hierarchical Feature Maps were an early attempt to cluster documents stories from a script [9]. The hierarchy was used to cluster these stories at three different levels which were the scripts (shop, rest, travel etc.), tracks (bus, plane, grocery etc.) and roles (J. Hamb, J. Fries, M. Spag etc). This formed a pyramid type representation, with each level clustering different information: the top level being the Scripts, the middle level the Tracks and Bottom level the Roles. This work has shown the potential uses of HSOM for interpreting information through visualisation and categories (clusters on the maps).

Work has also been done on hierarchical document clustering SOM in clustering a set of C++ help files, which were hierarchical by nature through inheritance of classes [10]. This work used a full set of words and a binary vector space representation of the documents. This means that if a word occurred it was marked as 1 in the document vector otherwise it remains 0. Then for any subsequent occurrence of this word in the same document and the vector still remains 1. Good results were obtained using this representation, which clearly underlined the hierarchical aspect of the C++ manual documents. Many advantages were also demonstrated such as the reduced computing in using the same set of documents the flat SOM took one hour whereas

the HSOM took roughly nine minutes [10]. This is because if a single map is used, a much larger map is needed and many divisions between data sets are also need (which is part of the HSOM branches). Another advantage of the HSOM is that when we move further down the hierarchy the number of documents and features (the vocabulary) of those documents are reduced, which also leads to a significant decrease in computing and vocabulary (set of terms) at each level.

4) Growing SOM

First we need to point out that there are two GSOM algorithms [11] and [12]. The later one is applied to the problem of knowledge discovery [12]. The difference with the growing grid (GG) [13] is that the GSOM is not restricted in inserting new rows or columns, meaning that it can have an arbitrary shape. This may be desirable for some applications. Like in the GG the GSOM has a growing phase and a fine tuning stage. The node-growing phase differs from the GG in that newly inserted nodes can be both within existing nodes or at boundary nodes (at the edges of the map). In GSOM the weight initialisation is more complex, but always uses the existing weight to perform interpolation or extrapolation as required for different cases. The growth stopping criteria is based on detecting a low number of newly added nodes. Finally the fine-tuning phase is actually called the smoothing phase for the GSOM whose purpose is to reduce the Quantisation Error. This method was found to have a flexible structure useful in knowledge discovery.

There are also other variants algorithms which could be used for document clustering such as the Growing Cell Structures (GCS) [14], the Neural Gas Algorithm [15], Incremental Grid Growing [16] and recently iSOM [17] and ViSOM [18][19].

5) Growing Hierarchical SOM

More recently the Growing Hierarchical SOM (GH-SOM) has recently introduced for document clustering [20]. This combines both the growing and hierarchical SOM methods to organise documents into a set of hierarchically based clusters. This method uses the GG algorithm [13] for the growth of the SOM. This is done by inserting rows or columns between the node with the largest number of BMUs (after a fixed number of adaptation steps) and the neighbour with the most different reference vector. The previously described HSOM was used to form the hierarchy in this method [10]. However the GH-SOM method uses a decreasing neighbourhood and learning rate as opposed to the GG in which they are fixed during growth. This method has the advantage of requiring considerably less computing that a flat structure can be set to a required level of detail (controlled by growth thresholds) and can easily be used to explore and visualise complex multidimensional data hierarchically.

IV. PROPOSED METHOD AND RESULTS

In the proposed method we followed the VSM pre-processing method matrix [1] with the "stop list", plural stemming and weighting scheme in equation (1), to create a representative vector for each document in the collection.

We then selected features from the frequency of occurrence of a term in each document, using a upper and lower thresholds, 5% and 85% respectively. Other statistical methods are also possible such as Latent Semantic Indexing [21] or Random Projection [8]. Once the indexing was finished, the document vectors were normalised. This is required otherwise the longer documents might have an advantage over the shorter documents, as more terms may

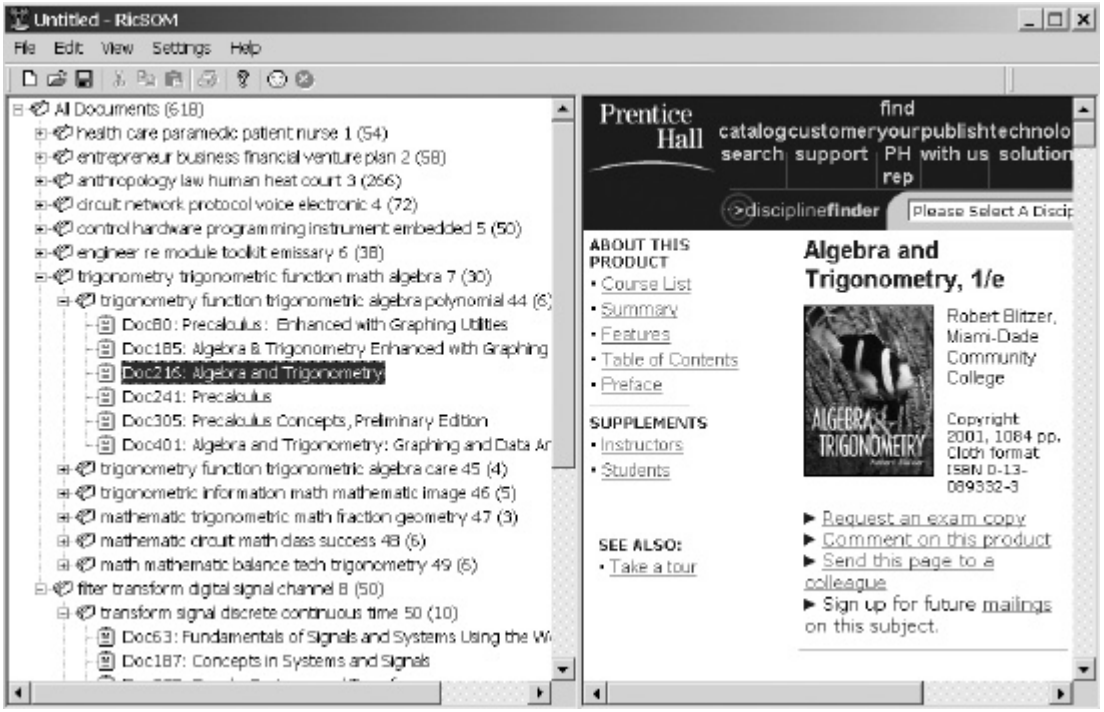

Figure 1 – View of the "1 + 1" tree organisation and document explorer application

occur in larger documents.

The proposed method essentially uses a series of independently spun, dynamically growing one-dimensional SOM, that are allocated hierarchically to organise the given set of documents. The advantage of using one-dimensional hierarchical structured, "1 + 1" type maps, is that the produced maps can very easily be visualised as a hierarchical tree as in Fig. (1). This is an *intuitive* representation for navigating topics or organising document and can be found in many areas from libraries (with the widespread Dewey Decimal Classification) up to the files and directory structures used in a Graphical User Interface application (such as Windows Explorer and Silicon graphics File Manager). This is a very different view from the methods used in the GH-SOM [20], were the output results are a set of two-dimensional maps. A two-dimensional representation makes a large set of documents confusing and difficult to navigate as the map become larger. In using a one–dimensional set of maps in "1 + 1" structure the navigation is simpler as we can easily backtrack to a previous topic, view many topics, expand and collapse any of the tree branches using the "+" and "-" symbols. Opening a document is also simple, as when a document is selected in the tree in the left window it is automatically opened and displayed in the Right window.

For labelling the directories of topics (clusters) we are currently using the LabelSOM technique [22]. The documents contained inside each directory are used to discover which words are best suited for labelling that particular directory. This was done using the Quantisation Error (QE) for each term within a cluster. The QE normally gives better results than relying on the frequency of occurrence of terms in a cluster, as it is the same values that the SOM uses to organise and cluster the documents. The QE was calculated using the Euclidean distance between the vectors of words and weights. These words were then ranked using this value. The first five of these are displayed as a label for each of the topics or directory in the hierarchy. The QE is accurate since it is the error measure used by the SOM to decide on the Best Matching Unit and when the growth should terminate. Other alternatives for labelling could have been used such as the method used in the WEBSOM [23]. In the WEBSOM method the main idea is to choose words, which are prominent in the cluster to be labelled and less prominent in the rest of the collection. Having the directories automatically labelled further enhances the user's navigation and understanding.

The proposed method differs from the algorithm used to grow maps in the GG [13]. In the GG algorithm, new nodes are only inserted as rows or columns of nodes within the existing nodes in the map, which are initialised using interpolation. This differs from the approach taken in the G-SOM method growth is performed by adding new inserting new nodes between existing nodes or at boundary nodes [12]. These node's weights were initialised using either extrapolation or interpolation methods. The approach to map growth in the proposed method uses both growth at boundary nodes and growth between existing nodes. This is different from the GH-SOM that grows each map using the GG: only growing between existing nodes.

To evaluate the similarity of a document and the weight of a node, the Manhattan distance (sum of the differences) was chosen, as it is more computationally efficient than the Euclidean and cosine correlation. Also all three are reported giving similar results for clustering documents [24]. Hence we used the Manhattan distance to find the BMU. The weights were then updated, using the same equation (3) previously described when introducing the SOM. The same neighbourhood equation (4) was also used.

In the proposed method each map is initialised with two nodes except for the topmost root map (we call Level 0) in the hierarchy that starts with three nodes. Two nodes are a good start as growth in a one-dimensional map can be made from both boundaries (using extrapolation) and between the existing nodes (using interpolation). The growth terminating condition was defined as a QE threshold. This threshold can be set lower producing larger and more detailed topic directories, or higher making smaller and broader topic directories. During growth the learning rate is fixed as in [13]. Once the growth is finished, the fine-tuning of the weights is performed by both decreasing the neighbourhood and learning rate as in the basic SOM previously described.

We shall now present the data sets and views of the generated tree for each set, using the proposed method. The three web document sets are descriptions of books available for purchase. Their content ranges from only a title to full table of contents, introductions, features, summaries, related course lists and one to three pages of text. These sets of realistic documents are good choices for testing the efficiency of the proposed method. Please note that due to the lack of space the fully expanded trees are not shown.

Data Set A – This is the "old" (July 2001) set of high school book descriptions from the Addison Wesley (www.aw.com), which is of variable content and length (one page to six pages). This set comprises of 269 documents (3.96Mb excluding images). Fig. 2 shows the generated tree hierarchy using the proposed method for Data Set A. Looking solely at the labels (and not the document titles) we can clearly understand the main dominant topics in each directory. At the lowest level we can clearly distinguish between the categories of economics & market, policy and trade, corporate & managerial finance, mathematics, calculus, chemistry & astronomy, anatomy & physiology and biology & genetics. Furthermore in the expanded branch it can be clearly seen that the directory "videotape human anatomy" is clearly correctly labelled and corresponds to the document titles.

Data Set B – This set consists of 618 web pages from Prentice Hall (www.prenhall.com), the size of document excluding images is of 20.0 Mb. These text documents are of variable content and length (one page to eight pages). Fig. 3 shows the generated tree on hierarchy Data Set B. At the lowest level of this tree the following main themes can be observed: healthcare & paramedics, business & financial,

anthropology & law, circuits & network protocols, hardware & programming, engineering, maths & trigonometry and filters & signals. Two directories are opened to show the difference between two neighbouring clusters. The first labels show a web & modern entrepreneur directory, while the second shows documents helping to succeed as a business and entrepreneur.

Figure 2 – Tree View for Data Set A

Data Set C – This set is the "new" (November 2001) set of high school book descriptions from the Addison Wesley (www.aw.com), which is of variable content and length (one page to twelve pages). Data Set C is a set of 1260 web pages and the size of document excluding images is of 29Mb. This set is also much more technical in content with a large number of documents about computers, software and programming.

Fig. 4 shows the generated tree view for the Data Set 3, in which we can distinguish the categories of algebra & trigonometry, algorithms & programming, enterprise & trade programming, software engineering & CASE (Computer Aided Software Engineering), networks & protocols, web & commerce, graphics software and market & economics. In the directory "privacy network policy cryptography encryption" we can easily find the documents dealing with security documents. The titles of the documents further verify this.

Figure 3 – Tree View for Data Set B

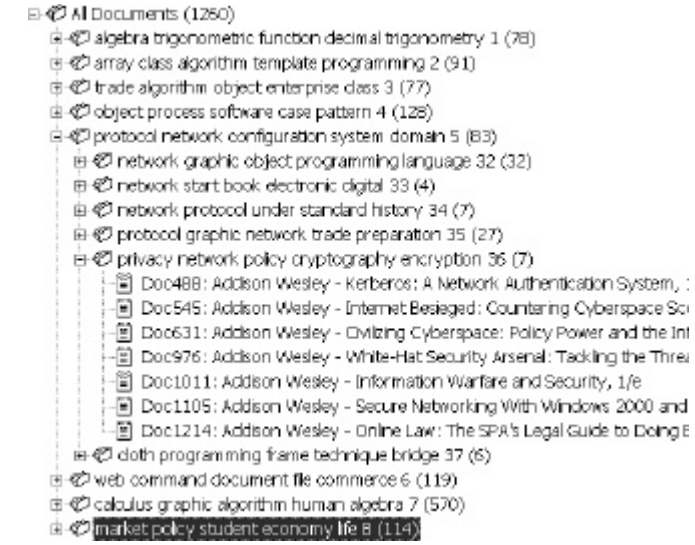

Figure 4 – Tree View for Data Set C

Richard Freeman, Hujun Yin and Nigel M. Allinson, "Self-Organising Maps for Tree View Based Hierarchical Document Clustering", in Proceedings of the IEEE IJCNN'02, Honolulu, Hawaii, 12-17 May, 2002. vol. 2, pp. 1906-1911.

For more details and enhancements please refer to the journal papers listed on http://www.rfreeman.net/

V. CONCLUSION

We have overviewed the area of document clustering using SOMs. Then we have proposed a method that allows a set of independent spun growing SOM maps organised in a hierarchy, to be directly used as a tree view of document topics. These topics represent key words, for which the majority of documents have in common in a particular cluster. This method has shown a more intuitive representation of a set of clustered documents, understanding the underlying topics and navigating a set of document.

In future work we shall be looking more closely at methods to enhance feature selection of words such as LSI [21] or Random projection [8]. We shall also look at ways to improve the accuracy of the directory labelling.

REFERENCES

[1] G. Salton, *Automatic text processing : the transformation, analysis, and retrieval of information by Computer*, Reading, Mass.Wokingham : Addison-Wesley 1988.

[2] M. F. Porter, An algorithm for suffix stripping, Originally published in *Program*, 14 no3, pp 130-137, July 1980.

[3] B. Everitt, *Cluster analysis*, 3rd ed. - London: Edward Arnold, 1993.

[4] T. Kohonen, *Self Organizing Maps Second Edition*, Springer 1997

[5] S. Haykin – *Second Edition Neural Networks a comprehensive Foundation* – Printice-Hall, 1999.

[6] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, Exploration of full-text databases with self-organizing maps. In Proceedings of the ICNN96, *International Conference on Neural Networks*, volume I, pages 56-61. IEEE Service Center, Piscataway, NJ.

[7] T. Kohonen, Exploration of very large databases by self-organizing maps. In Proceedings of ICNN'97, *International Conference on Neural Networks*, pages PL1-PL6. IEEE Service Center, Piscataway, NJ.

[8] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, V. Paatero, and A. Saarela. Self Organization of a Massive Document Collection. IEEE Transactions on Neural Networks, *Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, volume 11, number 3, pages 574-585. May 2000.

[9] R. Miikkulainen, Script recognition with hierarchical feature maps. Connection Science, 2(1&2):83--101, 1990.

[10] D. Merkl, 1997, Exploration of text collections with hierarchical feature maps, In: *Proceedings of the Int'l ACM SIGIR Conference on R&D in Information Retrieval*, Philadelphia, PA, 186–195.

[11] H.-U. Bauer, T. Villmann, Growing a Hypercubical Output Space in a Self-Organizing Feature Map, *ICSI Tech Rep*. TR-95-030 (1995).

[12] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan, Dynamic self organizing maps with controlled growth for knowledge discovery, *IEEE Transactions on Neural Networks*, vol. 11, pp. 601–614, 2000.

[13] B. Fritzke. Growing Grid: A self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 2(5), 1995.

[14] B. Fritzke, Growing Cell Structures - a self-organizing network in k dimensions, *Artificial Neural Networks 2* 1992.

[15] T.M. Martinetz, S.G. Berkovich, K.J. Schulten, "Neural -Gas" Network for Vector Quantization and its Application to Time-Series Prediction, *IEEE Transactions on Neural Networks*, Vol. 4, No. 4, July 1993, pp. 558-569.

[16] J. Blackmore and R. Miikkulainen. Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map. *In Proc of the IEEE Int'l Conf on Neural Networks (ICNN'93)*, San Francisco, CA, USA, 1993.

[17] H. Yin and N.M. Allinson, " Interpolating self-organising maps (iSOM)," *Electronics Letters*, Vol. 35, No. 19, pp. 1649-1650, 1999.

[18] H. Yin, " Visualisation induced SOM (ViSOM)," in *Advances in Self-Organising Maps*, N. Allinson, H. Yin, L. Allinson, J. Slack (Eds.), Springer, 2001, pp. 81-88.

[19] H. Yin, "ViSOM - A novel method for multivariate data projection and structure visualisation," to appear in *IEEE Trans. on Neural Networks*, Vol. 13, No. 1, 2002.

[20] M. Dittenbach, D. Merkl, and A. Rauber (2000): The Growing Hierarchical Self-Organizing Map, *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*.

[21] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, & R. Harshman, Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 391-407, 1990.

[22] A. Rauber, LabelSOM: On the labelling of selforganizing maps. In Proc. *International Joint Conference on Neural Networks*, Washington, DC, 1999.

[23] K. Lagus, and S. Kaski,. Keyword selection method for characterizing text document maps. Proceedings of *ICANN99, Ninth International Conference on Artificial Neural Networks volume 1*, pages 371-376, IEE, London.

[24] W.-C. Wong and A. Fu, Incremental Document Clustering for Web Page Classification, *IEEE 2000 Int. Conf. on Info. Society in the 21st century: emerging technologies and new challenges (IS2000)*, Nov 5-8, 2000, Japan.