Justification of HLabelSOM: Automatic Labelling of Self Organising Maps

Hiong Sen Tan
School of Computer and Information Science
University of South Australia
Adelaide, Australia
tan@cs.unisa.edu.au

Susan E. George
School of Computer and Information Science
University of South Australia
Adelaide, Australia
susan.george@unisa.edu.au

Abstract

HLabelSOM is a novel method to automatically label self organising maps. In this paper, we justify the method by defining the criteria of a good map. We first define the criteria of a good map for information retrieval, and then we justify the HLabelSOM for these criteria. We use medical documents retrieved from the web as experiment data to show the applicability of the HLabelSOM method. We also discuss other automatic labelling methods, LabelSOM and Lin's method, as comparison. Finally, we show that HLabelSOM outperforms other methods in producing good maps for information retrieval.

1. Introduction

In information retrieval, the common way to find documents is by entering keywords and hoping the system 'understands' the words we use in order to return the documents we expect relevant to our need. The key to success in this information retrieval mechanism is in the formulation of the keywords; this can be a problem for the users who are not very familiar with the domain.

One of the solutions to this keyword selection problem is to eliminate the necessity to provide a keyword in the information retrieval mechanism. We can give information about the document collection to the users and let them browse and explore, based on the information given, to get the documents they need. In this information retrieval mechanism, the users do not need to formulate the keywords as the keys are given.

The clustered map of self organising map (SOM) has been used on many occasions to retrieve the documents [1-4]. But, until now the information retrieval using a map is still not as popular as information retrieval by entering the keywords. One of the problems is due to the map itself, whether it is a good map in terms of being easy to understand, easy to use, and functional in its intended purpose as a browsing and exploration tool for information retrieval.

The SOM un-doubtedly can produce a good map. In fact, it can produce an accurate and ordered map. Several measures, such as quantisation error and topology preservation measures [5-6], justify the goodness of the map produced by the SOM technique. However, is the accuracy and the ordering, justified by the measures, of the map that the users are concerned about in information retrieval system? If it is not, what are the criteria of a good map from the users' points of view?

In this paper, we first define four properties a map should hold in order the map can be said 'good' for information retrieval. Then, we propose a novel method, HLabelSOM, to label automatically the SOM and last we justify that the HLabelSOM method is able to fulfil the criteria of a good map. We also discuss other automatic labelling methods, LabelSOM [2,7] and Lin's method [1], as comparison.

2. Document encoding

A document can be represented by a weighted vector or binary vector. The common way to weight the terms in the document is by using a frequency-based weighting function, known as inverse document frequency (IDF). Since not all the terms, e.g. 'a', 'the', and 'and', in the document are good for indexing purpose, the indexing process chooses only certain terms to be the keywords of the document. The IDF function then can be define as:

$$IDF_{ik} = \frac{\text{Frequency of keyword } k \text{ in document } i}{\text{Frequency of keyword } k \text{ in all documents}} \quad (1)$$

Using the IDF, a document D_i is represented by a vector [IDF_{i1}, IDF_{i2}, …, IDF_{iK}], where K is the number of keywords.

In contrast, the frequencies of the keywords are ignored in the binary vector. A document D_i will be represented by vector [0|1, 0|1, … , 0|1]. The given column is 1 if keyword k occurs in the document, and 0 otherwise. The binary presentation of the documents can only form a certain maximum number of patterns P, that is $P_{max} = 2^K$ or $P_{max} = 2^K - 1$ if we eliminate the zero vector [0,0, …, 0]. For example, if we have $K = 3$ and k_1 = ’a’, k_2 = ’b’, and k_3 = ’c’ then $P_{max} = 7$. All of them are P_1 = [‘a’], P_2 = [‘b’], P_3 = [‘c’], P_4 = [‘a’, ‘b’], P_5 = [‘a’, ‘c’], P_6 = [‘b’, ‘c’] and P_7 = [‘a’, ‘b’, ‘c’]. Also, $P_1 \subseteq P_1$, $P_1 \subseteq P_4$, $P_1 \subseteq P_5$, $P_1 \subseteq P_7$ and so on. We also can say if we have a document with an explicit pattern P_4 (for example), then implicitly we also have documents with the subsets of P_4, those are P_1, P_2, and P_4.

3. Exact match and approximate match

The subset relation of keywords plays an important role in information retrieval based on keywords. If we are looking for documents that have keywords ‘a’ and ‘b’, we expect the information retrieval system will return the documents with keywords ‘a’ and ‘b’, and we also do not mind if the system gives documents with keywords ‘a’, ‘b’, and ‘c’. In our system, we call the former as an exact match and the later as an approximate match. It is an exact match if the hamming distance of two vectors, vector A represents the document and vector B represents the searched keywords, is equal to 0. The formula is as follows:

$$D_H(A,B) = \sum_{i=1}^{K} \|A_i - B_i\| \tag{2}$$

where K is the number of keyword.

To calculate the approximate match, the hamming distance is modified as follows:

$$D_{AH}(A,B) = \sum_{i=1}^{K} \begin{matrix} 1, \text{ if } B_i > A_i \\ 0, \text{otherwise} \end{matrix} \tag{3}$$

The approximate match is achieved if the distance is equal to 0. In HLabelSOM, the approximate match is used to map the documents to the nodes on the map. The approximate match function covers the exact match function as well since if it is an exact match then it is an approximate match.

4. Experiment data

The experiment data are 40 medical documents fetched from the web. After the indexing process using automatic indexing software, we found total 308 keywords in all the documents. We reduce the number of keywords by choosing only 12 keywords that have the most occurrences in all documents. Then, each document is transformed into a 12-element-keyword of binary vector. The list of keywords used can be found in table 1 column 1. The number of explicit patterns formed is 31.

5. The self organising map

The SOM is an artificial neural network that is able to perform data clustering. One of the most important aspects of the SOM is that it is primarily a visualisation method for the clustering. Unlike statistical methods or other ways of grouping data the SOM provides a topological ordering where the relationships between data items are made apparent. In fact, relationships between data of high-dimensionality are reduced to relationships on a two-dimensional surface.

The SOM has a two-dimensional matrix of processing elements (the output layer) and a scalar array of processing elements (the input layer). Each node in the output layer is connected to every node in the input layer by weights. A training process is necessary to change the initially random weights into values that respond to input data. Please refer to [8] for detail of the SOM and [9] for the practical version of the algorithm.

The two-dimensional output map effectively orders the input data once the training is complete. To make the ordering and the clusters apparent to the users we need to label the map. We can use a visualisation technique, such as U-matrix [10], cluster connection [11] and adaptive coordinate [12], or we can use an automatic labelling method, such as HLabelSOM that we propose in this paper, LabelSOM [2,7] and Lin’s method [1].

6. The goodness of the map

The map is said to be good if the outputs represent accurately the inputs and the map has been in order. The accuracy can be measured by a quantisation error function as follows:

$$E_Q = \frac{1}{N} \sum_{i=1}^{N} \|x_i - m_c(x_i)\| \tag{4}$$

where N is the number of inputs and $m_c(x_i)$ is the output that have the closest distance from the input x_i.

The ordering of the map is how well the map keeps the relations in the input. It can be verified by using any topology preservation measures, such as topographic error [5] or a measure proposed by Kaski and Lagus [6].

7. Criteria of good map for information retrieval system

However, in information retrieval systems that use map display as a search tool to browse and explore the document collection, users will see the goodness of the map from different points of view. The users will be more concerned about whether the maps are easy to use, easy to understand, and function as their intended purpose rather than whether the maps are accurate and in order, justified by the measures mentioned above.

The map will help the users if after it is labelled, it holds these following properties:

Table 1. Keywords and their frequencies.

Keyword	F_{doc}	Frequency on 4 x 4 map				
		F_{eq}	F_{min}	F_{max}	F for λ =	
					0.50	0.41–0.45
arthritis	10	4.0	4	4	4	4
asthma	9	3.6	3	4	4	4
cancer	9	3.6	3	4	3	3
care	8	3.2	3	4	3	3
cause	11	4.4	4	5	5	5
diabetes	10	4.0	4	4	4	4
disease	12	4.8	4	5	5	5
health	11	4.4	4	5	4	4
joint	8	3.2	3	4	3	3
pain	8	3.2	3	4	2	3
patient	9	3.6	3	4	4	5
treatment	12	4.4	4	5	6	6
DE =					0.17	0.17
DD =					0.48	0.48

7.1. Continuity

A topology preservation measure guarantees that the map is in order. Therefore, regardless the labelling method used, the method should keep this continuity. What 'continuity' means here is that 'similar documents should be located close each other. The neighbouring nodes should be labelled by same patterns of keywords or share the common keywords to form clusters'. If the map does not hold this continuity, the map will be meaningless for browsing.

7.2. More and richer patterns on the map

The quantisation error measures how accurately the outputs represent the inputs. We probably cannot have all the explicit patterns in the input present on the map because the capability of showing all explicit patterns is limited by the map size (see the map size discussion below). But, of course we would like to have map(s) with more patterns and more keywords in the patterns, since the more patterns and the richer the keywords the patterns have, the easier it is for users to find specific documents. It is also possible that the map forms a zero pattern [0, 0, …, 0] and new patterns that are not present in the input, even though the new patterns are probably non-sense in some applications. For example, in animal domain a new pattern with keywords '4_legs' and 'fly' is a joke.

7.3. The equally keyword distribution

Given a labelled map, we probably conclude that if a keyword has higher or lower frequency than those of other keywords on the map then this keyword should have higher or lower frequency than those of other keywords in all documents. This is based on an equal distribution mechanism. The user will understand quite easily what the collection is about if the keyword frequencies in the input give equal distribution to the keyword frequencies in the output.

7.4. The smaller map size

We can show all the explicit patterns of the inputs by using a particular map size corresponding to the number of the input patterns. For example, if we have 31 explicit patterns of input, we probably can use 6 x 6 map (36 nodes) or bigger to make sure that all explicit patterns present in the output. But, it is not always a good solution since the space, i.e. computer screen resolution, is limited. And, even though we can use the scrolling function to move the screen window, we still prefer to see the smaller area as our working space. One of the solutions is to use hierarchical maps to reduce the working space.

In the next sections we will examine how the automatic labelling methods fulfil these criteria. In general, the automatic labelling method at least should hold the continuity property otherwise the map is meaningless. The more patterns and the richer the keywords the patterns have, the easier it is for users to find specific documents. The later two, the equally keyword distribution and the smaller map size, are not as essential as the formers, but they make the map easier to understand and use.

8. Automatic labelling

What we mean by automatic labelling of SOM here is a way to label to the nodes based on the weight values the nodes hold after the training. In this section, we will discuss HLabelSOM, the method we propose, LabelSOM and Lin's method.

8.1. HLabelSOM

Previously in HLabelSOM [13], we labelled a node with a keyword if the node weight value for a particular keyword is greater or equal to 0.5. The selection of value 0.5 as a threshold is based on the fact that the document vectors are binary vectors. If the final weight value is above or equal to 0.5, the node should have the keyword, otherwise it should not.

But, we still should justify whether 0.5 is the appropriate value for the threshold. If it is not, then what is the appropriate value? As a simple example, if we have input 0 and 1 and we train 1 x 1 SOM, what do we expect? We expect that the final weight is 0.5, but it isn't always the case, at least with the algorithm that we use, the final weight values are not exactly 0.5, but about 0.5. There are several parameters of the SOM trainings influence the final weight, such as learning rate, neighbourhood function, and input sequences. We need an appropriate threshold that the value of it is between 0 and 1, and most probably about 0.5.

Since we label the node by comparing each weight, i.e. the weight of the keyword in each node, the selection of the threshold should have direct effect to the keyword distribution. Based on this fact, we should try to find a value of the threshold that makes the labelled map mirrors the keyword distribution most equally.

We apply the above idea into our experiment data. For example, if keyword 'arthritis' occurs 10 times in 40 documents, it should appear 4 times on 4 x 4 map (16 nodes) in order it can be said that it is equally distributed. If keyword 'asthma' occurs 9 times in 40 documents, it should appear 3.6 times (3 or 4 times) in 4 x 4 map. The complete frequencies of keywords in whole documents (F_{doc}), the equally distributed frequencies of keywords on the map (F_{eq}), minimum (F_{min}) and maximum (F_{max}) frequencies that are allowed, are given in table 1 column 2 to 5.

After the SOM is trained (and measured by some measures mentioned above to choose a 'good' map), the selection of the threshold value is undertaken before the HLabelSOM labelling method is applied. The distribution error (DE) values, which are the numbers of keywords that are not equally distributed, are calculated for all possible threshold values (λ) from 0.00 to 1.00. The DE is calculated and normalised as follows:

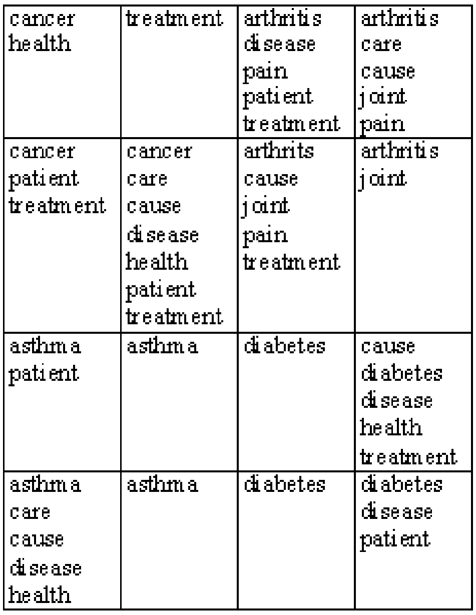
$$DE = \frac{1}{K}\sum_{i=1}^{K} \begin{array}{l} 0, \text{ if } F_{\min} <= F <= F_{\max} \\ 1, \text{ otherwise} \end{array} \tag{5}$$

where K is the number of keywords,

To extent the preciseness of the distribution error, we calculate distribution distance (DD) that is the distance from F to F_{eq}, as follows:

$$DD = \frac{1}{K}\sum_{i=1}^{K} \left\| F - F_{eq} \right\| \tag{6}$$

Now, as we have DE and DD values for all possible threshold values between 0.00 and 1.00, we can choose the threshold value that most equally distributed the keywords, which is the threshold value with minimum DE (DE_{min}) or minimum DD (DD_{min}). For our experiment data, we found DE_{min} = 0.17 and DD_{min} = 0.48 for λ = 0.41, 0.42, 0.43, 0.44, 0,45 and 0.50. In this case, we choose the lowest λ, i.e. 0.41, since a low threshold increases the frequencies of the keywords that means more keywords in the patterns (see section 7.2 for the 'more and richer patterns on the map' discussion). The distributions of keyword frequencies for the above thresholds are shown in table 1 column 6 and 7. The result map of applying HlabelSOM with λ = 0.41 is shown in figure 1.



cancer health	treatment	arthritis disease pain patient treatment	arthritis care cause joint pain
cancer patient treatment	cancer care cause disease health patient treatment	arthrits cause joint pain treatment	arthritis joint
asthma patient	asthma	diabetes	cause diabetes disease health treatment
asthma care cause disease health	asthma	diabetes	diabetes disease patient

Figure 1. The detail map. The map is labelled by using HLabelSOM with λ = 0.41.

As we refer to our novel method as the HLabelSOM, where H stands for Hierarchical, our purpose is to have a hierarchical visualisation. We believe in creating user interface that follows the information visualisation mantra: "overview first, zoom and filter, then details on demand" [14] can lead to an effective information retrieval system. A variation of the SOM, the HSOM [15, 16], will be able to achieve this purpose as well. But, using HSOM will require several trainings for different SOMs. We can also produce hierarchical maps by combining four adjacent nodes on the map into one node and naming the new node with the common keywords the four nodes share if the common keywords exist, otherwise with all keywords the four nodes have [17]. The absence of the common keywords in the four adjacent nodes leads to a drawback of this method because renaming a new node with all keywords that the four nodes have does not really represent the documents the node contains.

In the HLabelSOM method, the map is re-labelled to produce other maps. In the new maps the nodes are labelled with the keywords if the node weight value for the particular keyword is greater or equal to λ and only the n greatest weight values of keywords are selected. We can choose the values of n from 1 to a certain value that will lead to the production of the detail map (figure 1) as a result. Figure 2 shows the labelled map for λ = 0.41 and n = 1, this is an overview map. Since we have the overview map and the detail map now, we can always make intermediate maps between them, so called middle maps.

Figure 3 shows the middle map for λ = 0.41 and n = 2. We can now use the maps in figure 1, figure 3, and figure 2 as hierarchical maps: the detail map, more general (middle) map and most general (overview) map.

8.2. LabelSOM

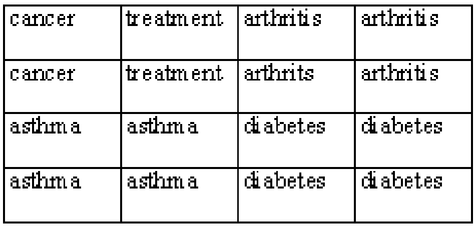
LabelSOM [2,7] labels the nodes on the map based on the quantisation errors of all keywords that are accumulated distance between the weight vector elements of all input mapped to the nodes. The quantisation errors that are close to 0 or below a given threshold (λ_1) indicate the keywords best characterise the input mapped to the nodes. Further, we need another threshold, especially for the applications where we find a high number of input vectors that have values of 0, e.g. text documents. These zero values mean the keywords are not present in the particular input. The selected keywords should have weight vectors above the second threshold (λ_2). Figure 4 shows the result map when LabelSOM is used with λ_1=0.10 and λ_2 = 0.50.

For binary input vectors, we found out that the node can be labelled as follows: the node is labelled by all keywords of the document input if only one input is mapped to the node, and by the common keywords of the document inputs if more than one input mapped to the node. In this case, we do not need to calculate the accumulated quantisation error and define the thresholds.

8.3. Lin's method

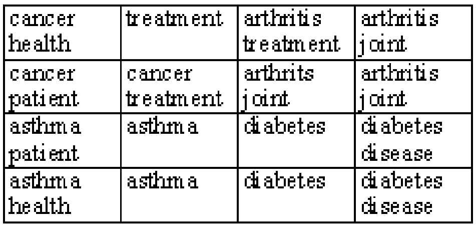
Lin's method [1] labels the nodes by comparing each node to all unit vectors, which are vectors consisting of only one keyword, and labelling the node with the name of the winning keyword, which is a unit vector that has the closest distance to the node. As a result the areas on the map are continuous, shown in figure 2. The Lin's method gives the same result as HLabelSOM with n = 1 does.

The most important thing in the three automatic labelling methods above is the differences in the time and the method used of mapping the documents to the nodes on the map. In HLabelSOM, the mapping of the documents to the nodes is done after the map is labelled and the documents are mapped to the nodes for approximate matches. In LabelSOM, the mapping of the documents to the nodes is done before the map is labelled and the documents are mapped to the nodes that have minimum distances from the documents. In Lin's method, the documents are mapped to the nodes that have minimum distances as in LabelSOM, but the mapping can be done before or after the labelling since the inputs do not have any contribution to the labelling mechanism.



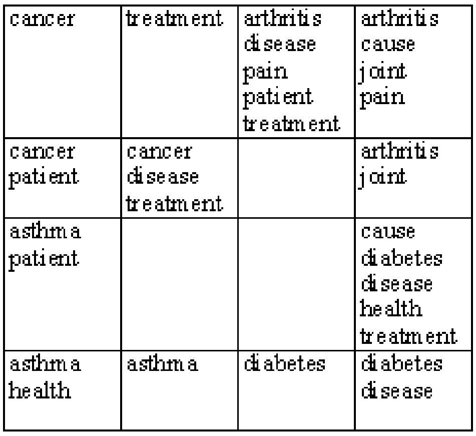
cancer	treatment	arthritis	arthritis
cancer	treatment	arthrits	arthritis
asthma	asthma	diabetes	diabetes
asthma	asthma	diabetes	diabetes

Figure 2. The overview map. The map is labelled by using HLabelSOM with λ = 0.41 and n = 1. Using Lin's method also produces this map.



cancer health	treatment	arthritis treatment	arthritis joint
cancer patient	cancer treatment	arthrits joint	arthritis joint
asthma patient	asthma	diabetes	diabetes disease
asthma health	asthma	diabetes	diabetes disease

Figure 3. The middle map. The map is labelled by using HLabelSOM with λ = 0.41 and n = 2.



cancer	treatment	arthritis disease pain patient treatment	arthritis cause joint pain
cancer patient	cancer disease treatment		arthritis joint
asthma patient			cause diabetes disease health treatment
asthma health	asthma	diabetes	diabetes disease

Figure 4. The map is labelled by using LabelSOM

9. Conclusion

We conclude the automatic labelling methods in respect to the properties of the good information retrieval map, which we defined above, should hold.

HLabelSOM, LabelSOM and Lin's method show the continuity on the map, neighbouring nodes are labelled by the same keywords or share the common keywords to form the clusters. The users will be able to utilise this continuity property to browse and explore the map to find the documents they need.

HLabelSOM, when more than one map is used, certainly has maps with more patterns and richer keywords in the patterns than LabelSOM and Lin's method do. The more patterns and the richer the keywords the patterns have, the easier it is for the users to find specific documents.

The selection of the threshold value used in HLabelSOM will lead to the production of the map towards the equally keyword distribution. We do not see any attempt to equally distribute the frequencies of the keywords in the input to the output in LabelSOM and Lin's method, as seen in table 2. The equal distribution of the keywords will give more clue to the users what the collection is about.

We do not need to worry about the map size by using the HLabelSOM because we can always stack the map hierarchically, to present more patterns, instead of making the map size bigger.

Now, we can say that the HLabelSOM outperforms the two other automatic labelling methods, LabelSOM and Lin's method, in respect to the four properties of a good information retrieval map. But, as we mentioned before, the continuity property, that the LabelSOM and Lin's method also posses, is sufficient for the map to be used for browsing and exploration in information retrieval. The LabelSOM is the most accurate method to map documents onto the nodes since the mapping is based on the quantisation errors of the inputs mapped to the nodes. Lin's method is always useful to give the overview of the collection without much detail and shows the continuity and the classification more clearly than other methods do since it labels the nodes with minimum number of keywords (mostly one keyword).

Table 2. Keyword frequency distribution on the 4 x 4 map using three different labelling methods.

Keyword	Frequency (*F*) on the map using		
	HlabelSOM	LabelSOM	Lin's method
arthritis	4	3	4
asthma	4	3	4
cancer	3	3	2
care	3	0	0
cause	5	2	0
diabetes	4	3	4
disease	5	2	0
health	4	2	0
joint	3	2	0
pain	2	2	0
patient	4	3	0
treatment	6	4	2
DE	0.17	0.67	0.75
DD	0.48	1.45	2.60

At the moment, to justify the wider applicability of the HLabelSOM method, we have been applying it onto a bigger collection of medical documents. The collection is about 6000 documents.

References

[1] X. Lin, D. Soergel and G. Marchionini. A self-organizing semantic map for information retrieval. Proc. of the 14th Annual Int'l ACM/SIGIR Conf. on Research and Development in Info. Retrieval, 1991.

[2] A. Rauber and D. Merkl. Automatic labeling of self-organizing maps for information retrieval. Journal of System Research Info. System, 10 (10): p.23-45, 2001.

[3] T. Kohonen et al. Self organization of a massive document collection. IEEE Trans. on Neural Networks, 11(3), 2000.

[4] A. L. Houston et al. Medical data mining on the internet: research on a cancer information system. Artificial Intelligence Review, 13: p. 437-466, 1999.

[5] K. Kiviluoto. Topology preservation in self-organizing maps. IEEE Conf. on Neural Networks, p.294-299, 1996.

[6] S. Kaski and K. Lagus. Comparing self-organizing maps. Proc of the Int'l Conf. of Artificial Neural Networks (ICANN96), 1996.

[7] A. Rauber. LabelSOM: on the labeling of self-organizing maps. Proc. of the Int'l Joint Conference on Neural Networks (IJCNN'99), Washington, 1999.

[8] T. Kohonen. Self-organizing map. 2nd Ed. Springer-Verlag, 1995.

[9] R. P. Lippmann. An introduction to computing with neural nets. IEEE Acoustics, Speech, and Signal Processing Society, p. 4-22, 1987.

[10] A. Ultsch. Self-Organizing neural networks for visualisation and classification. Information and Classification. Concepts, Methods and Applications. Springer: Berlin, 1993.

[11] D. Merkl and A. Rauber. Cluster connections: a visualzation technique to reveal cluster boundaries in self-organizing maps. Proc. of 9th Italian Workshop on Neural Nets (WIRN97), Vietri sul Mare, Italy, 1997.

[12] D. Merkl and A. Rauber. On the similarity of eagles, hawks, and cows: visualization of semantic similarity in Self-Organizing Maps. Proc. of Int'l Workshop Fuzzy-Neuro-Systems'97, Soest, Germany, 1997.

[13] H. S. Tan. HLabelSOM: Automatic Labelling of Self Organising Maps toward Hierarchical Visualisation for Information Retrieval. 16th Australian Joint Conference on Artificial Intelligence (AI'03), Perth, Australia, 2003.

[14] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. Proc. of IEEE symposium on visual languages. Boulder, CO, USA, 1996.

[15] R. Miikkulainen. Script recognition with hierarchical feature maps. Connection Science, 2 (1) p.83-102, 1990.

[16] D. Merkl. Exploration of text collections with hierarchical feature maps. Proc. of the 20th Int'l ACM SIGIR Conf. on Research and Development in Info. Retrieval. 1997.

[17] H. S. Tan and S. E. George. Multi-media based web mining for an information resource. 3rd Int'l Conf. on Data Mining Methods and Databases for Engineering, Finance and Other Fields (Data Mining 2002), Bologna, Italy, 2002.