A Constructive and Hierarchical Self-Organising Model in A Non-Stationary Environment

Chihli Hung
Computational Intelligence Group
De Lin Institute of Technology, Taiwan
chihli@mail.educities.edu.tw

Stefan Wermter
School of Computing and Technology
University of Sunderland, UK
stefan.wermter@sunderland.ac.uk

Abstract –**Several related self-organising neural models have been proposed to enhance the flexibility of self-organising maps (SOM). These models are focused on the ability of continuous learning in a non-stationary environment. In our studies, these models depend on the pre-definition of several thresholds which are used as guidance of neural behaviours for specific data sets. However, it is not trivial to determine those thresholds in a non-stationary environment. When a proper threshold has been determined, this threshold may not be suitable for the future. Therefore, in this paper, we compare the dynamic adaptive self-organising hybrid (DASH) model with the growing neural gas (GNG) model by introducing several different initial thresholds to test their feasibility. Our experiments show that the DASH model is more stable and practicable for document clustering in a non-stationary environment since DASH adjusts its behaviour not only by modifying its parameters but also by an adaptive structure.**

I. INTRODUCTION

In the era of information overload, too much irrelevant information overwhelms the user. Based on the concept of *cluster hypothesis* [17], grouping documents with similar concepts for information access reduces the search space and provides more meaningful clusters for users. Clustering organises information, and thus is becoming more important. In the field of artificial neural networks, self-organising maps (SOM) have been proposed for clustering [11]. The SOM model is usually designed for a static data collection. It is trained by a static training set and tested by an unseen test set. Thus, the model can generalise well under the assumption that the unseen test set is similar to the training set.

However, the real world information is continuously growing and often changes over time, which means that the boundary of the unseen test set is hard to be defined and therefore the unseen test set is usually different from the training set. For example, in a news collection, some specific events occur over a specific period. In other words, the news topic is changing over time. A training set composed by the older news events is not always appropriate to represent the news events. Thus the particular neural structure of the current static document collection that is learned and identified by the SOM model may be outdated for new information.

Many SOM-like approaches have been proposed for a non-stationary clustering task, for example, growing neural gas [5], growing neural gas with utility criterion [6], the grow-when-required technique [15] and the dynamic adaptive hybrid self-organising model [10]. These models usually contain unit-growing and unit-pruning functions which are analogous to biological functions of remembering and forgetting under a non-stationary environment. These models also depend on the pre-definition of several thresholds which are used as guidance of neural behaviours for specific data sets.

However, it is not trivial to determine those thresholds in a non-stationary environment. A set of better parameters often requires several iterations of trial and error or rules of thumb from experience [8]. Even though a proper threshold has been found, this threshold may not be suitable for the future in a non-stationary environment.

On the other hand, if a clustering model is able to tackle novelty in a non-stationary environment, the model still needs to face the difficulty of a large quantity of information, which is hard to analyse and demonstrate efficiently. Therefore, breaking the problem into smaller pieces is one policy to analyse and solve the complex task [1, 14].

Like traditional statistical agglomerative hierarchical algorithms [4], the artificial neural learning models are also able to cover hierarchical clusters. The dynamic models with a unit-pruning function have the ability to provide such a structure. When the model is relatively stable, by observing the sequence of pruning behaviour, the earlier the class of units is separated, the higher level it is in a hierarchy [e.g. 7]. However, a model with a unit-pruning function is not guaranteed to form one or more separate structures, because this pruning behaviour is data-oriented.

One alternative way is that a hierarchical structure is offered by recursive processing from a unit or an area which has a high accumulated error or many input vectors mapped to it [e.g. 16, 3]. Therefore, a map at a deeper level of a hierarchy shows finer clusters from a unit or an area which contains too much error or too many associated input samples. However, these hierarchical models are designed for a stationary environment only.

In this paper, we compare the dynamic adaptive hybrid self-organising (DASH) model with the growing neural gas (GNG) model by introducing several different initial thresholds to test their feasibility. A detailed analysis of setting parameters for two models is discussed. We show that the DASH model is a constructive and hierarchical self-organising model, which is more stable and practicable for document clustering in a non-stationary environment.

The remainder of this paper is organised as follows. In Section 2, we define the non-stationary environment for news documents. In Section 3, we compare GNG with DASH models based on their algorithms. In Sections 4 and 5, we discuss influence of parameters for GNG and DASH models respectively. In section 6, we introduce the hierarchical feature of DASH. Finally, we give a conclusion in section 7.

II. A NON-STATIONARY ENVIRONMENT FOR NEWS DOCUMENTS

This paper examines the feasibility of the GNG and DASH models in a non-stationary environment where the existing data set is treated as outdated knowledge and is updated by new knowledge, i.e. the new data set. This can occur when a neural clustering model continues to learn more than one document set from different periods of time. The main aim of our models is to represent the latest clustering structure in a non-stationary environment. Since time is a crucial element for news articles, different documents from different time periods contain different related stories, which may use different words or phrases.

In the Reuters-RCV1[1] news corpus, a collection of 10,000 full-text news articles is treated as one data set. The first 10,000 full-text news articles are called the existing data set and the second 10,000 full-text news articles are called the new data set. We focus on news articles in the eight most dominant topics of the Reuters-RCV1, use only open-classed words and then remove words from a stop list defined in WordNet[2]. Each data set contains about five days of news articles, 1,000,000 word occurrences, 16,000 distinct words and 78.28% co-occurring words.

It is necessary for text processing to transform text to vectors based on a vector representation approach. The traditional vector space model (VSM) [19] is used to represent a full-text document in this paper, since VSM does not involve human classification knowledge and it is one of the best-known text vector representation approaches. However, this method is likely to suffer from *the curse of dimensionality* because the dimensionality of the document-word matrix is the total number of different words in the VSM model. Similar to the work of Chen et al. [3], only the 1,000 most frequent words from the master word list are used in our experiments since this method has provided the greatest overlap in representations [18] and has been shown to be as good as most dimensionality reduction techniques [20, 2].

To mimic a non-stationary environment, we train a constructive neural model using the existing data set and introduce the new data set at iteration 30,000. Thus, a constructive neural model should learn from experience and adapt itself to the new data set.

III. GROWING NEURAL GAS VS. DYNAMIC ADAPTIVE HYBRID SELF-ORGANISING

In this section, we compare DASH with GNG from the algorithmic viewpoint. Details of the DASH and GNG algorithms can be found in [9, 5] respectively. We consider both models an extension of Kohonen's self-organising map (SOM). SOM uses a pre-defined topological structure of units and a time-decaying learning rate such that adjacent units contain similar weights and therefore units self-organise into an ordered map [11]. However, SOM is not suitable in a non-stationary environment because the learning is stopped after the learning rate reaches a very small value. Furthermore, it is hard to presuppose the inner structure of a large and non-stationary data set, so such a pre-defined SOM topology is not appropriate for a dynamically changing environment.

To extend the practicability of SOM, growing neural gas (GNG) is proposed [5]. GNG starts with two units and connects an input vector's best matching unit (BMU) to the second best matching unit (SMU). The BMU is an output unit which contains the shortest Euclidean distance to the current input sample. At each iteration, each age variable of all connections that are directly linked to the BMU is increased but the age variable of the connection between BMU and SMU is initialised to zero. This behaviour uses a local function since it is not necessary to consider whole connections on the map. After every pre-defined period, a new unit is inserted by splitting the unit with the highest error in the direct neighbourhood from the unit with the highest error in the whole structure. The unit-pruning function removes old connections whose age variables exceed a pre-defined threshold. Thus isolated units which have no connections are removed. However, an unsuitable pre-defined threshold for unit-pruning may prevent the model from growing or pruning at all.

The dynamic adaptive hybrid self-organising (DASH) model not only adapts its architecture but also its main parameters to input samples. Like GNG, the DASH model starts with two units and one connection. Unlike GNG, the age variables of all connections except the one between BMU and SMU are increased at each iteration. This is a quasi-global function since it considers all connections in the current map for the current data set to remove outdated connections between units. Please note that GNG prunes the unsuitable units learned from the current data set by removing isolated units, which are formed by trimming connections whose age variables are greater than a pre-defined threshold. However, the age variable of a connection is incremented only because one of its associated units is the BMU. Thus, some connections may contain a low-age variable because their associated units are not activated often. Consequently, some unsuitable units are kept not because these units still represent the current input samples but simply because these units are not activated often. Therefore, we argue that this

[1] Reuters RCV1 news corpus can be found at http://about.reuters.com/researchandstandards/corpus.

[2] WordNet can be found at http://www.cogsci.princeton.edu/~wn/

local function used in GNG is sub-optimal in a non-stationary environment.

In terms of learning in a non-stationary environment, a fixed learning rate and pre-defined neural structure may be impractical. Both GNG and DASH use two fixed learning rates and a dynamic growing structure to adapt per se to a non-stationary environment. However, some pre-defined unit-pruning and unit-growing thresholds of neural models such as GNG are also hard to presume in a non-stationary environment. An unsuitable unit-growing threshold affects the training length while an unsuitable unit-pruning threshold may make the model never grow. As the aim of learning in a non-stationary environment is continuous learning [10], the learning length seems not to be a key factor. However, if a constructive neural model cannot grow, this model is hard to learn. Thus, DASH is equipped a self-adjusting function, which mainly adjusts its unit-pruning threshold according to the experience of the previous training.

Finally, DASH is a hierarchical self-organising model while GNG is a flat model, which represents all input vectors using only one map. Inspired by some hierarchical self-organising models, such as M-SOM [3] and GHSOM [16], DASH develops a sub-map based on the average quantization error set by an information analyst. The average quantization error (AQE) is suggested by Kohonen as a measurement used in the vector quantization technique [12]. The quantization error, also called the *distortion measure*, is defined as the sum of the Euclidean distance between every input vector and its Best Matching Unit (BMU). The AQE is the average value of quantization error to the number of input vectors. It is an indicator of the quality of the model or the unit, which represents an average input sample. Thus the DASH model uses the top map to represent the latest data set and develops a sub-map when a unit of the parent map cannot represent the associated input samples well, i.e. a unit with a high AQE. The hierarchical growing procedure of DASH is illustrated in Fig. 1 and the hierarchical structure of DASH is illustrated in Fig. 2.

```
 1. Setting global objective AQE
     2. Training for a map.
         3. Learning phase.
         4. Pruning phase.
         5. Growing phase.
         6. If AQE of a map meets the objective AQE of this local map, then
            skip step 7.
         7. Self-adjusting phase and going to step 3.
         8. Put extra training of this map in the global training pool if the AQE
            of any unit in this map does not meet the objective AQE.
     9. If the global criteria have not met, then go to step 2.
10. End.
```

Fig. 1. The hierarchical growing procedure of DASH

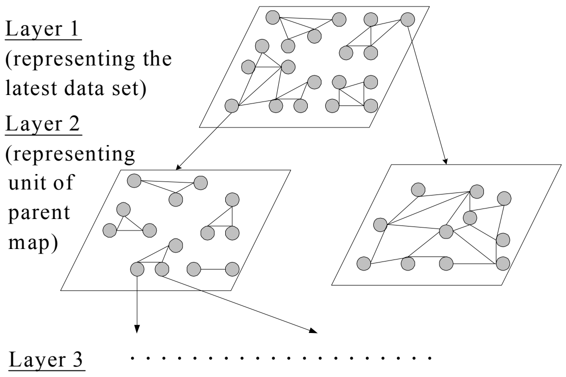

Layer 1
(representing the latest data set)
Layer 2
(representing unit of parent map)
Layer 3
. . . .

Fig. 2. The hierarchical structure of DASH

IV. PARAMETERS FOR GNG

There are four main parameters for GNG. The first is a pre-defined learning rate, ε_b , for the best matching unit (BMU). The second is a pre-defined learning rate, ε_n , for all direct neighbours of the BMU. Learning rates decide how far output units move toward input samples. The third is a pre-defined unit-growing cycle, which decides how often GNG grows. The fourth is a pre-defined unit-pruning threshold, β , which determines the threshold of outdated connections. All connections whose age is greater than β will be trimmed and units without any connection will be pruned.

In this paper, we only focus on the pre-defined unit-pruning threshold because an unsuitable unit-pruning threshold may prevent a constructive neural model from growing and this parameter is also hard to pre-define in a non-stationary environment. Other parameters do affect the training length if the stop criterion is based on the quality of the model, such as AQE.

TABLE I
A COMPARISON OF GNG MODELS WITH DIFFERENT β

β	ε_b	ε_n	growing cycle	number of units	AQE
105	0.1	0.001	313	148	0.859
90	0.1	0.001	313	148	0.862
75	0.1	0.001	313	148	0.859
60	0.1	0.001	313	146	0.861
45	0.1	0.001	313	143	0.860
30	0.1	0.001	313	140	0.867
15	0.1	0.001	313	104	0.876
10	0.1	0.001	313	41	0.927
5	0.1	0.001	313	2	0.999
0	0.1	0.001	313	2	0.996

In our experiments, the existing data set is used in the beginning and the new data set is introduced at iteration 30,000. We use the same parameters except the unit-pruning threshold, β , for different experiments as in Table I. This parameter is difficult to determine when learning in a non-stationary environment since we may have no opportunity to find a proper value for a new data set. The value of AQE for all GNGs is decreasing in general during training before iteration 30,000 (see Fig. 3). When the new data set is introduced at iteration 30,000, the AQE is back to a very high value as in Fig. 3. This situation illustrates that the output clusters based on the existing data set are not able to represent those input samples from the new data set. Thus, we expect that many output units learned from the existing data set should be pruned in order to learn from the new data set. On the other hand, when β is under 30, the number of output units for GNG is under 140 (see Fig. 3 and 4). That is, a very small unit-pruning threshold will prevent GNG from growing.

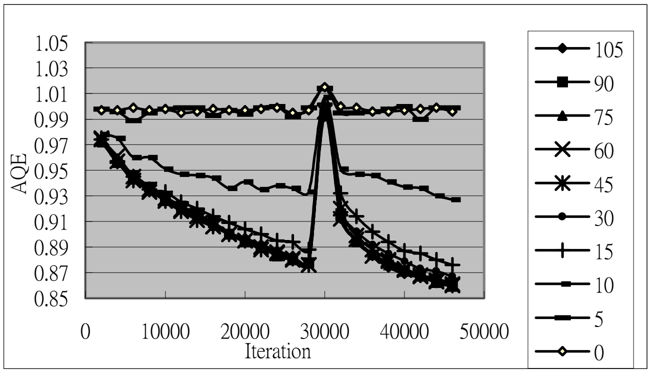


Fig. 3. A comparison of AQE for GNG models with different β

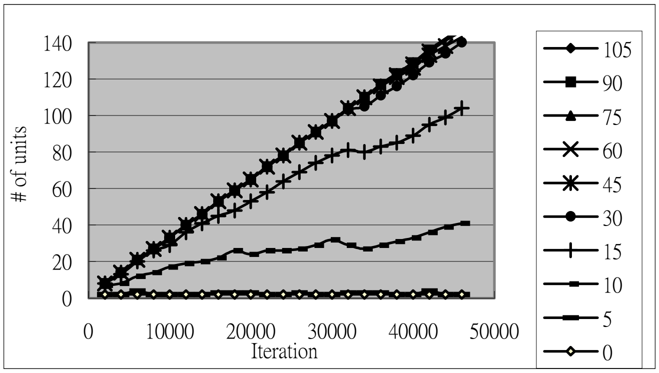


Fig. 4. A comparison of unit usage for GNG models with different β

V. PARAMETERS FOR DASH

Compared with GNG, DASH needs two more parameters, which are stop criteria, i.e. the τ and S_{min}. In the beginning, we use a unit which is the mean vector of current samples to represent current samples in a training map. Thus we have an AQE as:

$$AQE_0 = \frac{1}{N}\sum_{i=1}^{N}\left\|x_i - w_0\right\|, \qquad (1)$$

where N is the number of current input samples in the model and w_0 is the mean vector of current input samples. An information analyst can define an objective map quality index, τ, for a child map. That is, the aim of the training is to reduce the AQE in the upper layer to $\tau \times$ AQE. Thus, the τ parameter influences the training length: the smaller the value of τ, the more time is needed to train a specific map. This parameter also influences the size of the DASH map because the training length relates to the size of a map for incremental growing neural networks in general. The other criterion for DASH training is S_{min}, which is defined as the minimum number of input documents for the next recursive training cycle. That is, the further sub-map is not developed for a unit if the number of its associated input documents is less than S_{min} no matter what value of AQE the unit has.

The other four parameters are the same as used in GNG. In this section, we still focus on the unit-pruning parameter, β, since others do not prevent DASH from growing. Due to the nature of the dynamic structure of the DASH model, it is helpful to understand the influence of the unit-pruning parameter to the constructive behaviour of DASH. The β variable is self-adjusting in DASH, which affects the connection-trimming threshold. To control the size of the map, the DASH model decreases the value of β during training in general but if the current β prevents DASH from growing, another slightly greater β will be used [10]. Thus, given an initial value of β, the DASH model will adapt to input samples during training. Theoretically, a large β reduces a small number of outdated connections and this large value is preferred when tackling unknown data since to remember is more import than to forget in the beginning of learning. Three DASH models with different β values are applied for comparison. The other parameters of these DASH models are the same.

According to the overall results in Table II, a small β needs a longer training time for the same objective AQE. Only the first 50,000 iterations are shown in Fig. 5-7 for comparison. The original β resumes when sub-maps are trained with different scales of input data. The AQE is decreasing during training but gets back to a very high value when the new data set is introduced at iteration 30,000 (Fig. 5). At the mean time, as the AQE still exceeds the objective AQE, the value of β should be reduced to adapt to the new data set. Thus, DASH removes several unsuitable units at iteration 30,000 (see Fig. 6 and 7). However, they finally have similar AQE (see Table II). In summary, according to this study, the initial value of β does not cause the DASH model to continue training indefinitely without learning. However, this situation may happen for models, such as GNG [5], GNG-U [6], GWR [15], etc. that depend on some constant thresholds to adjust its unit-growing or unit-pruning function if these thresholds are not set properly.

Table II
A COMPARISON OF DASH MODELS WITH DIFFERENT β

β	AQE for map1	AQE	training iteration for map1	unit # for map1	map #
95%	0.855	0.818	34,000	153	20
50%	0.858	0.820	152,000	124	17
25%	0.859	0.808	268,000	124	23

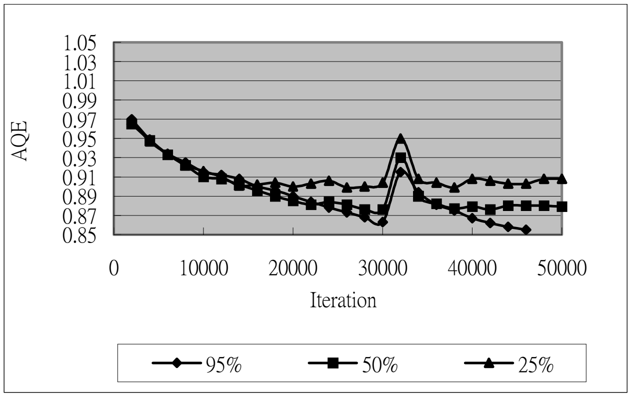


Fig. 5. A comparison of AQE for DASH models with different β

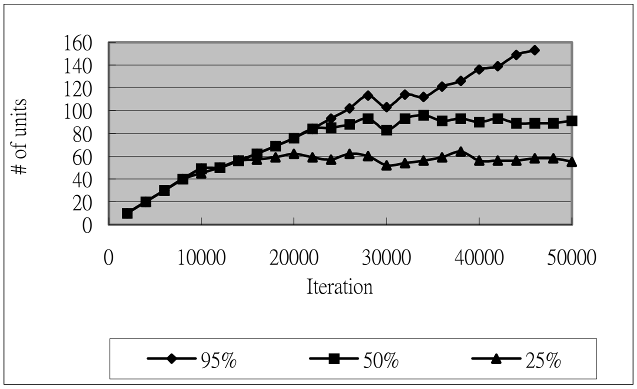


Fig. 6. A comparison of unit usage for DASH models with different β

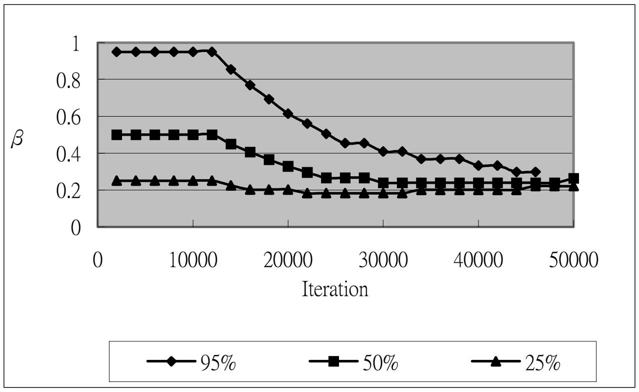


Fig. 7. A comparison of β variations for DASH models

VI. HIERARCHICAL KNOWLEDGE REPRESENTATION

One of the benefits of using the DASH model is to represent complicated documents in a hierarchical manner, which is considered useful for data analysis [14]. Theoretically, a SOM-like clustering technique identifies the natural groupings of documents in a high dimensional space by its self-organising units in a two-dimensional map. Thus, documents with similar concepts are clustered in the same unit and similar units are located nearby in such a space.

This approach provides a chance to evaluate DASH by a qualitative criterion, because the SOM is able to project input vectors from a multi-dimensional space to a two-dimensional space and keep the internal relationships among them faithfully [12]. In other words, if output units of DASH in a neighbourhood represent documents with similar concepts, these units will also be represented by the neighbouring SOM units.

To evaluate a SOM-like model by a qualitative criterion, it is necessary to assign meaningful labels to units. Roussinov and Chen [18] assign a term to each output unit of the SOM map by choosing the unit element that contains the largest value. This method has been used by several researchers [13, 3] and is also used in this paper. A square SOM map whose size corresponds to the number of DASH output units, is used to represent DASH results. Two terms out of 1,000 index words whose weights are the most significant are used to represent the labels of the unit in the first layer of the DASH hierarchy. The second and third significant terms are used to represent the map in the second layer. That is, the *n* and *n-1* most significant terms are used to represent the map in the *n* level of a hierarchy. Thus, a unit of the map in the lower layer of a hierarchy is associated with more terms, which represent news articles with more specific concepts.

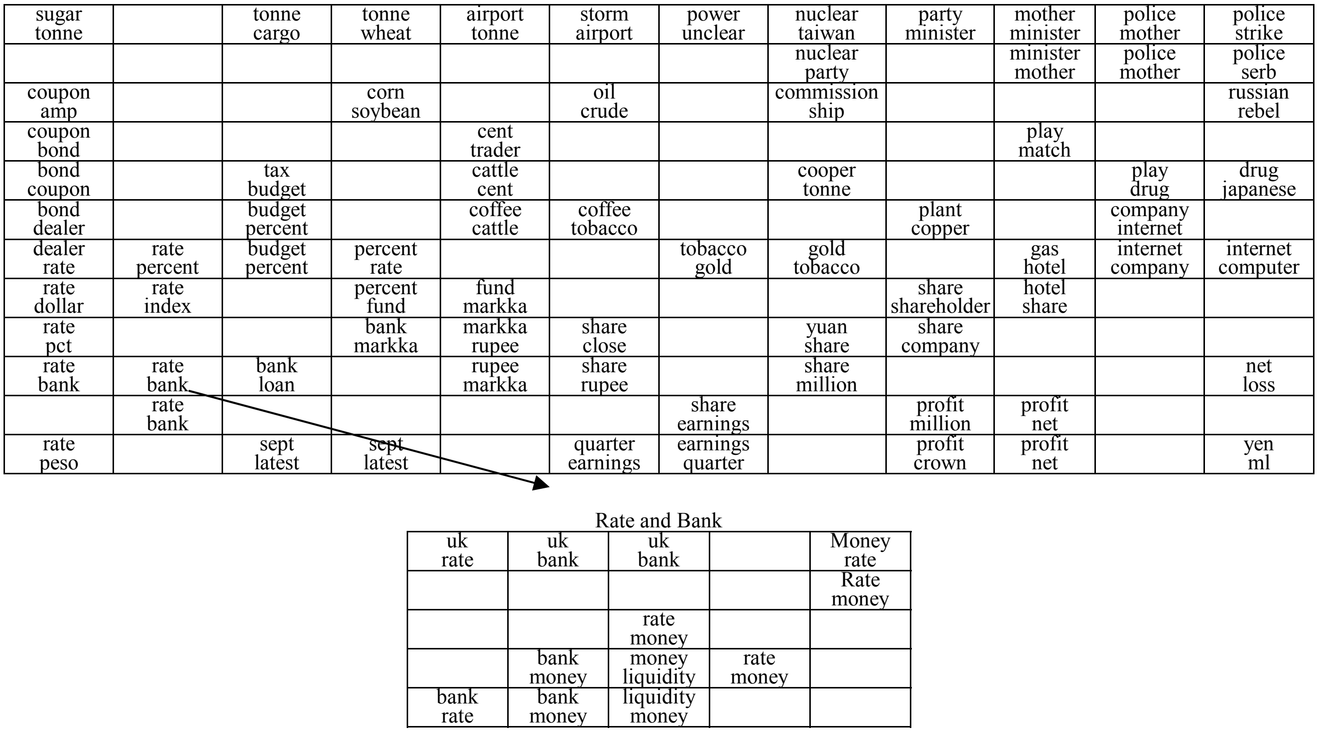
For illustration, the root-map and one of its sub-maps are shown in Fig. 8. Two neighbouring units in the DASH root map usually have one identical word or one related word, which demonstrates that units in a neighbourhood represent similar concepts and those concepts are altered smoothly. For example, units on the top left mainly discuss crops and trade, the units on the top right discuss administration issues, the bottom right units are related to company situations and the bottom left units discuss bank and money matters.

One unit labelled as “rate bank” further develops a 5x5 sub-map and the second and third significant terms are used to represent conceptual labels for its units (see Fig. 8). Units with similar concepts are placed nearby and these concepts are more specific than those in the root map.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have demonstrated that the DASH model overcomes the limitations of a self-organising model in a non-stationary environment. Compared with growing neural gas (GNG), the DASH model is more stable and feasible since it adapts to a new data set not only by the neural architecture but also by several crucial parameters. We also present the feature of the hierarchical training for DASH in order to further analyse the internal groupings of documents.

For the further research, we consider using DASH to handle further time-series data sets. The DASH model is able to track different document collections over time and keep updated document clusters. However, this is not a time-series text clustering task, since the word or document vector does not have an impact on other words or document vectors over time. We consider our DASH approach a time-based approach. The time series data set such as a financial time series can be seen as a quantitative index and the time-based document can be seen as a qualitative index. It could be useful to use knowledge from the DASH model to refer quantitative indices to qualitative indices and vice versa.



sugar tonne		tonne cargo	tonne wheat	airport tonne	storm airport	power unclear	nuclear taiwan	party minister	mother minister	police mother	police strike
							nuclear party		minister mother	police mother	police serb
coupon amp			corn soybean		oil crude		commission ship				russian rebel
coupon bond				cent trader					play match		
bond coupon		tax budget		cattle cent			cooper tonne			play drug	drug japanese
bond dealer		budget percent		coffee cattle	coffee tobacco			plant copper		company internet	
dealer rate	rate percent	budget percent	percent rate			tobacco gold	gold tobacco		gas hotel	internet company	internet computer
rate dollar	rate index		percent fund	fund markka				share shareholder	hotel share		
rate pct			bank markka	markka rupee	share close		yuan share	share company			
rate bank	rate bank	bank loan		rupee markka	share rupee		share million				net loss
	rate bank					share earnings		profit million	profit net		
rate peso		sept latest	sept latest		quarter earnings	earnings quarter		profit crown	profit net		yen ml

Rate and Bank

uk rate	uk bank	uk bank		Money rate
				Rate money
		rate money		
	bank money	money liquidity	rate money	
bank rate	bank money	liquidity money		

Fig. 8. Part of the hierarchical structure of DASH

REFERENCES

[1] M.J.A. Berry and G. Linoff, *Data Mining Techniques*. New York, Wiley, 1997.

[2] S. Chakrabarti, "Data mining for hypertext: a tutorial survey," *ACM SIGKDD Explorations*, vol. 1, no.2, 2000, pp. 1-11.

[3] H. Chen, C. Schuffels and R. Orwig, "Internet categorization and search: a self-organizing approach," *Journal of Visual Communication and Image Representation*, vol. 7, no. 1, 1996, pp. 88-102.

[4] W.R. Dillon, *Multivariate Analysis*, John Willey & Sons, Inc., 1984.

[5] B. Fritzke, "A growing neural gas network learns topologies," *Advances in Neural Information Processing Systems 7*, G. Tesauro, Touretzky, D.S. and Leen, T.K. eds., MIT Press, Cambridge MA, 1995, pp. 625-632.

[6] B. Fritzke, "A self-organizing network that can follow non-stationary distributions," *Proceedings of ICANN'97, International Conference on Artificial Neural Networks*, Springer, 1997, pp. 613-618.

[7] V.J. Hodge and J. Austin, "Hierarchical word clustering – automatic thesaurus generation," *Neurocomputing*, vol. 48, 2002, pp. 819-864.

[8] A.L. Hsu and S.K. Halgamuge, "Enhancement of topology preservation and hierarchical dynamic self-organising maps for data visualization," *International Journal of Approximate Reasoning*, vol. 32, no.2-3, 2003, pp. 259-279.

[9] C. Hung and S. Wermter, "A dynamic adaptive self-organising hybrid model for text clustering," *Proceedings of The Third IEEE International Conference on Data Mining (ICDM'03)*, Melbourne, USA, November, 2003, pp. 75-82.

[10] C. Hung and S. Wermter, "A time-based self-organising model for document clustering," *Proceedings of International Joint Conference on Neural Networks*, Budapest, Hungary, July, 2004, pp. 17-22.

[11] T. Kohonen, *Self-organization and associative memory*, Springer-Verlag, Berlin, 1984.

[12] T. Kohonen, *Self-Organizing Maps*, 3rd edition. Springer-Verlag, Berline, Heidelberg, New York, 2001.

[13] X. Lin, "Map displays for information retrieval," *Journal of the American Society for Information Science*, vol. 58, no. 1, 1997, pp.40-54.

[14] C.D. Manning and H. Schütze, H, *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, Massachusetts, London, 2002.

[15] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural Networks*, vol. 15, 2002, pp. 1041-1058.

[16] A. Rauber and D. Merkl, "Text mining in the SOMLib digital library system: the representation of topics and genres," *Applied Intelligence*, vol. 18, 2003, pp. 271-293.

[17] C.J. Van Rijsbergen, *Information Retrieval*, London, Butterworths, 2nd Edition, 1979.

[18] D.G. Roussinov and H. Chen, "Document clustering for electronic meetings: an experimental comparison of two techniques," *Decision Support Systems*, vol. 27, 1999, pp. 67-79.

[19] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*, Addison-Wesley, Reading, MA, 1989.

[20] H. Schütze and C. Silverstein, "A comparison of projections for efficient document clustering," *Proceedings of the 20th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, 1997, pp. 74-81.