A Time-Based Self-Organising Model for Document Clustering

Chihli Hung
De Lin Institute of Technology, Taiwan
and
University of Sunderland, UK
chihli.hung@sunderland.ac.uk

Stefan Wermter
Centre for Hybrid Intelligent Systems
School of Computing and Technology
University of Sunderland, UK
stefan.wermter@sunderland.ac.uk

***Abstract* – Most current approaches for document clustering do not consider the non-stationary feature of real world document collection. In this paper, in a non-stationary environment, we propose a new self-organising model, namely the dynamic adaptive self-organising hybrid (DASH) model. The DASH model runs continuously since the new document set is formed consecutively for training while the old document set is still at the training stage. Knowledge learned from the old data set is adjusted to reflect the new data set and therefore document clusters are up-to-date. We test the performance of our model using the Reuters-RCV1 news corpus and obtain promising results based on the criteria of classification accuracy and average quantization error.**

I. INTRODUCTION

Technological innovation has led to a rapid growth in the quantity of textual information. Such a wealth of information can overwhelm the user. By grouping similar sets of information, an organised document structure can reduce the search space and help users to access a number of related and potentially relevant documents [15].

Many document clustering approaches, including statistical solutions and artificial neural networks, have been proposed for these tasks. However, most of these approaches are based on a common assumption that the documents are organised as a stationary collection (i.e., a fixed number of documents). Thus, although the particular structure of the current stationary document collection has been identified, this becomes outdated for new information. Therefore, motivated by the need for non-stationary organisation of information, this document clustering project has been proposed.

The remainder of this paper is organised as follows. In Section 2, we define the stationary and non-stationary text models. In Section 3, we give a general description of self-organising neural clustering models. In Section 4, we review current related neural clustering models in a non-stationary environment. In section 5, we introduce the DASH approach. In section 6, we evaluate our proposed model using the new Reuters Corpus under different scenarios. A conclusion is presented in section 7.

II. STATIONARY AND NON-STATIONARY TEXT MODELS

Traditionally, the factor of time is not involved in an artificial learning environment for clustering. Documents, e.g. news articles, usually have some relationship with time. Similar articles related to the same specific event are presented in a specific time period. On the other hand, topics of news articles are gradually changed over time and the latest event generally attracts more attention. Therefore, the size of the document population is always increasing in a real world so the environment is non-stationary. The appearance of a new document that is not shown in the stationary document collection is inevitable, and is likely to produce a wrong decision for document clustering.

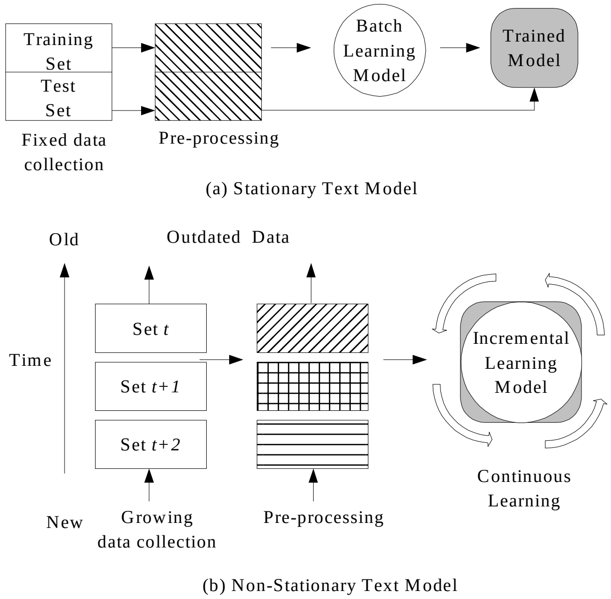
Most models which handle the task in a non-stationary environment are trained by introducing the input sample or the sub-set of samples one by one. This learning behaviour is more natural than batch learning in a non-stationary environment. Batch learning, which needs all input samples introduced to a model before learning, is impractical in a non-stationary environment since the new input samples do not yet exist and the large size of data is intractable for storage in computer memory.

The limitation of learning models in a non-stationary environment has been addressed by introducing the concept of non-batch learning, for example online learning, lifelong learning, incremental learning and knowledge transfer, which all identify the same limitation of using batch learning in a non-stationary environment but from different viewpoints [6]. Online learning stresses that some input samples are unavailable in a non-stationary environment since the total number of documents is unknown. Lifelong learning emphasises learning throughout the entire lifetime of the model, which should cope with a changing environment. Incremental learning trains a model with the new input sample, without wiping the old prototype sample. Knowledge transfer is a machine learning method which learns from one task to another task and takes advantage of previous training experience if the latter task is related to the previous one.

In a non-stationary environment, a text model should be able to learn continuously, keep up-to-date and provide the results at any time. Due to the nature of text processing which needs to transform each word or document to a vector based on the vector space representation approach, incremental learning by a sub-set instead of a single input sample is necessary. For training each sub-set, incremental learning by a single input sample can be used to offer the results at any training stage.

Thus, in terms of a non-stationary text model for news articles in this paper, an open data set is divided into several sub-sets based on different periods of time. The later sub-set is related to its previous sub-set but usually contains slightly different news events. The model always learns from the new sub-set and still preserves knowledge learned from the old sub-set if it does not contradict the current task. Therefore, the results (i.e. document clusters) of this non-stationary text model are gradually changing over time and always reflects the latest inner relationships of news documents grouped by concept.

In Fig. 1a, a fixed document collection is used for a text model in a stationary environment, which usually enforces batch learning. A new document that is not included in the original data collection (the training set + the test set) requires a re-training procedure to keep the model up-to-date. In contrast to a stationary text model, the non-stationary text model in Fig. 1b associates document time-stamps based on time. Each time-stamped packet contains several documents issued in the short period. The incremental learning model adapts to the new time-stamped data set by continuously adjusting the learned structure.



(a) Stationary Text Model

(b) Non-Stationary Text Model

Fig. 1. Stationary text model and non-stationary text model

III. SELF-ORGANISING NEURAL TEXT MODELS

Inspired by the biological concept, in which neurons with similar functions are placed together, Kohonen proposes a self-organising map (SOM) using a pre-defined topological structure of units and a time-decaying learning rate such that adjacent units contain similar weights, so units self-organise into an ordered map [8]. Therefore, users can choose the relevant clusters of documents on the map to get relevant documents. The robustness of the SOM algorithm and its appealing visualisation effects make it a prime candidate in neural text clustering.

However, a model that depends on the time-decaying learning rate is not suitable in a non-stationary information environment because the learning is stopped after the learning rate reaches a very small value. Furthermore, the network structure including the topology and the number of units has to be set before training. It is hard to presuppose the inner structure of a large and non-stationary data set, so such a pre-defined SOM topology may not be appropriate.

IV. RELATED SOM-LIKE MODELS IN A NON-STATIONARY ENVIRONMENT

Several related self-organising neural models have been proposed to enhance the practicability of SOM. A common goal of these algorithms is to map a data set from a high-dimensional space onto a low-dimensional space, and keep its inner structure as faithful as possible. These models are focused on the ability of continuous learning in a non-stationary environment. For example, the growing cell structure (GCS) [5], growing neural gas (GNG) [3], incremental grid growing (IGG) [1], growing neural gas with utility criterion (GNG-U) [4], plastic self organising map (PSOM) [10] and grow when required (GWR) [11], contain unit-growing and unit-pruning functions which are analogous to biological functions of remembering and forgetting under a non-stationary environment.

For the non-stationary data set, a trained unit or training unit should be updated by a unit which is trained with new input samples. This is performed by the unit-pruning or connection-trimming function. A model with the connection-trimming function should be based on a global age consideration. The reason is that a local age variable of a connection does not grow when units of this connection are not activated. That is, the aged connection may be kept forever so that the capability of self-adjustment for a model to new stimuli is diminished. Thus, a model, such as GNG and GWR, using the connection-trimming function based on a local age consideration can be treated as an incomplete non-stationary model.

On the other hand, an unsuitable constant unit-pruning or connection-trimming threshold may make the model train forever but learn nothing. This constant value can be very small or very large, which is totally dependent on trial-and-error. Therefore, it is not a good idea to use such a constant threshold for a big data set. Unfortunately, the GCS, GNG, IGG, GNG-U, PSOM and GWR apply a constant threshold for detection of unsuitable units. We argue that a unit-pruning or connection-trimming threshold should be automatically adjusted to suit different data sets during training.

V. DYNAMIC ADAPTIVE SELF-ORGANISING HYBRID MODEL

By inspecting limitations of existing neural text models, we propose the dynamic adaptive self-organising hybrid model (DASH). The complete DASH algorithm can be found in [7]. In this paper, we focus on the differences between this model and existing models. The DASH model adapts its main parameters and architecture to input samples in a non-stationary environment for document clustering. Learning in the DASH model is self-organised so units nearby represent similar documents. In terms of the concept, the DASH model is a hybrid integration of the growing neural gas (GNG) [3] and growing hierarchical self-organising map (GHSOM) models [14]. The flowchart of DASH is shown in Fig. 2, which involves two main iterations and seven processes. The inner iteration is a GNG-like learning procedure for each map in a hierarchy. The outer iteration is a GHSOM-like recursive training cycle.

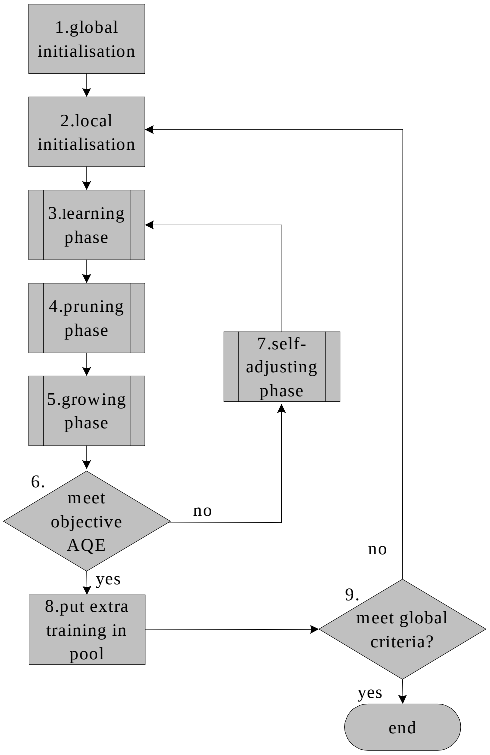


Fig. 2. The flowchart of the DASH algorithm

Like GNG, the DASH model starts with two units [Fig. 3] and applies the competitive Hebbian learning principle to connect the best matching unit and second best matching unit for an input stimulus [12]. A connection is trimmed if it is relatively old compared to other connections and a unit without any connection is removed. Like GHSOM, the recursive training continues for the individual unit whose average quantization error (AQE), which is the average distance between input vectors and their associated representative vector, is greater than the objective AQE.

Unlike GNG, the DASH model applies a global connection-trimming function instead of a local one for the current data set to remove outdated relationships between units. Therefore, the isolated units are pruned, which makes the DASH model able to forget outdated knowledge. The local function is sub-optimal, but is used by GNG since the global one does not exist in a non-stationary environment. However, a quasi-global function can be provided by transferring knowledge task by task in a non-stationary environment. Some work regarding knowledge transfer between neural networks can be found in the literature, e.g. [17, 13]. The concept of knowledge transfer is especially suitable for document clustering in a non-stationary environment. The reason for this is that document vectors for the artificial neural model are produced based on some vector representation approach which transforms relationships between words and documents to weight vectors.

Furthermore, this connection-trimming threshold is a constant for GNG but it is a self-adjusted variable based on input vectors for the DASH model. This threshold is increased if units are not growing in the DASH model Eq. (1) and is decreased if the number of units has reached the reference number of units in a map Eq. (2). The reference number of units is a temporal maximum unit number for the current map and is also increased when this number is reached Eq. (3). GNG grows every pre-defined constant cycle which is determined by trial-and-error. In contrast, this cycle is a part of the DASH model, which is mutually decided by the objective AQE and the number of input samples in the current map. Finally, GNG is a flat model, which represents all input vectors using a map, but the DASH model is a hierarchical model, which is able to train a whole input set gradually by training several smaller input sub-sets separately. The GNG-like growing behaviour of the DASH model is illustrated in Fig. 3a-3f.

$$\beta(t+1) = \beta(t) \times (2 - J_{\beta}), \tag{1}$$

where ß is a connection age threshold, *t* indicates time, $J_{\text{ß}}$ is the ß adjusting parameter which is between 0.5 and 1.

$$\beta(t+1) = \beta(t) \times J_{\beta}\,. \tag{2}$$

$$O_l(t+1) = O_l(t) \times (2 - J_O), \tag{3}$$

where O_l is the reference number of units in a map and J_O is its adjusting parameter which is between 0.5 and 1.

The main difference between the DASH model and GHSOM is that GHSOM is designed for a stationary environment but the DASH model can be used in a non-stationary environment. This is because GHSOM is based on the traditional SOM training algorithm, which contains a time-based decaying learning rate. Even though the stop criterion for the DASH model and GHSOM is the quality of clustering (i.e. the AQE), GHSOM still needs to decide the training length for each static SOM training. GHSOM is a recursive Growing Grid (GG) model: another GHSOM is created from the unit in its parent GHSOM. However, without a unit-pruning function, once units grow, they are part of the GHSOM architecture forever. Unlike the GHSOM model, the DASH model is a recursive GNG model that contains both unit-growing and unit-pruning functions, which offer the elasticity for a competitive learning model used in a non-stationary environment. The GHSOM-like hierarchical training of the DASH model is illustrated in Fig. 3g and 3h.

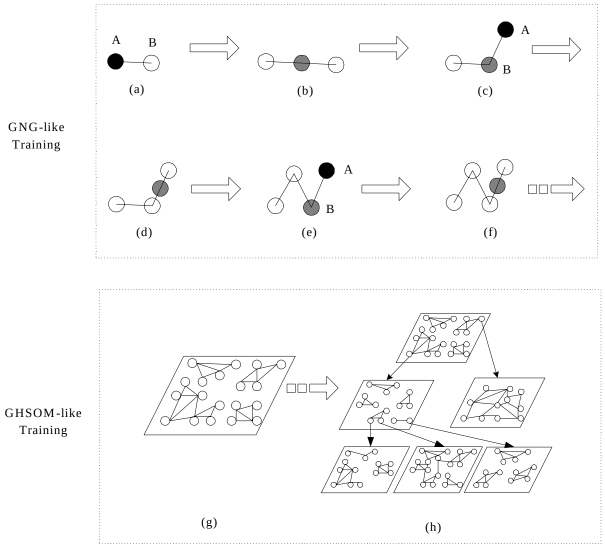


Fig. 3. The growing processes for the DASH model. Units are represented as circles and lateral connections between units are represented as lines. Circle A indicates the unit with the biggest error and Circle B indicates the neighbour with the biggest error for Circle A. The grey circle is the new unit at each stage.

VI. EXPERIMENTS

A. Experiment Design

Two data sets are used for the simulation of a non-stationary environment. For convenience, the first one is called the existing data set and the second one is called the new data set. The current version of the Reuters news corpus, RCV1, is used in this research and the eight most prominent topics are focused on. Since a news article can be pre-classified as more than one topic, we consider the multi-topic as a new combination of topics. Thus, the 8 chosen topics are expanded into 40 combined topics for the first 10,000 news articles.

We treat the first 10,000 full-text news articles in Reuters-RCV1 as the existing data set and treat the following 10,000 full-text news articles as the new data set. Each data set uses a normalised TFxIDF vector representation method [16] based on its own period of time. In other words, they form different word-document matrices which are based on the first and the second 10,000 full-text documents for the existing data set and new data set respectively.

The existing data set is used for all scenarios in the beginning and the new data set is introduced in scenario 1 at iteration 10,000, scenario 2 at iteration 30,000 and scenario 3 at iteration 50,000. In other words, the existing data set is updated by the new data set at iteration 10,000 for scenario 1, iteration 30,000 for scenario 2 and iteration 50,000 for scenario 3. For convenience, the first scenario is termed iter10000, the second scenario is termed iter30000 and the third scenario is termed iter50000.

We use open-class words, i.e. nouns, verbs, adjectives and adverbs, remove the stop words, and lemmatise each word to its base form. We further pick up the 1,000 most frequent words from the master word list since this method is as good as most dimensionality reduction techniques [2].

We compare the DASH model with SOM and GNG because these two models are typical models in the static neural clustering group and dynamic neural clustering group respectively. For comparison with the DASH model, the same training length, learning rate and the number of units used in the first layer of the DASH model should be used for SOM and GNG. In our experiments, the objective AQE is 90%, the initial connection-trimming threshold (i.e. *ß*) is 95%, the *ß* adjusting parameter (i.e. $J_ß$) is 90% and the unit reference number adjusting parameter (i.e. J_O) is 90% for the DASH model. Under these settings, the DASH models need 42,000, 46,000 and 62,000 iterations for scenario 1, scenario 2 and scenario 3 respectively. Thus, these training lengths are also used for SOM and GNG.

B. Evaluation

We evaluate our model by AQE and classification accuracy, which have also been used in the work of Kohonen et al. [9]. AQE and classification accuracy for each scenario are shown in Fig. 4 and 5. According to these results, the overall DASH hierarchy outperforms other models with a higher classification accuracy and a lower AQE for all scenarios.

According to Fig. 4, SOM suffers from a non-stationary environment when the new data set is introduced at a later training stage. In scenario 1, which uses 42,000 iterations and introduces the new data set at iteration 10,000, SOM gradually adjusts itself to suit the new data set during the rest training length (i.e. 32,000 iterations). The performance evaluated by classification accuracy criterion is even slightly better than other dynamic models (i.e. the DASH model and GNG). However, in scenarios 2 and 3, the rest training lengths are 16,000 (46,000-30,000) and 12,000 (62,000-50,000) iterations respectively for training the new data set. SOM performs worse than the DASH model since its learning rate has decayed to a small value. In other words, the training length in scenarios 2 and 3 is not enough for SOM to keep the same performance in scenario 1. A comparison of models based on AQE criterion is illustrated in Fig. 5. AQE of SOM is 0.937, 0.948 and 0.956 in scenario 1, 2 and 3 respectively. SOM has the worst results in all scenarios. This effect is more evident when the new data set is introduced at a later training stage.

GNG also suffers from a non-stationary environment because a new data set that is introduced at a later training stage produces a greater AQE and lower accuracy [Fig. 4 and 5]. Unlike SOM and GNG, the DASH model does not suffer from introducing the new data set at a later training stage. Classification accuracy for the first layer of the DASH hierarchy is 65.36%, 66.16% and 65.77% and AQE is 0.856, 0.855 and 0.857 in scenario 1, 2 and 3 respectively [Fig. 4 and 5].

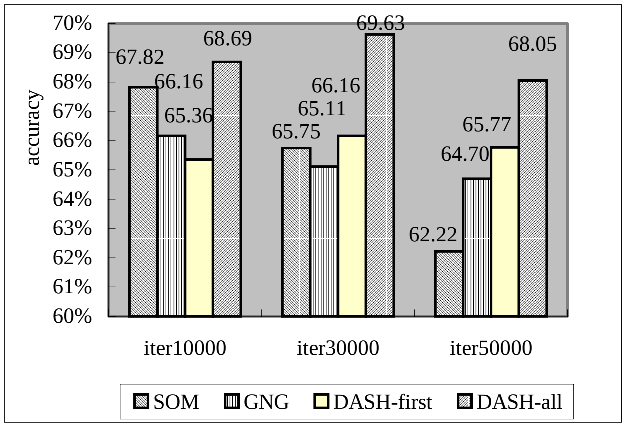


Fig. 4. A comparison of SOM, GNG and DASH evaluated by classification accuracy. DASH-first denotes the first layer of the DASH hierarchy and DASH-all denotes the whole DASH hierarchy.

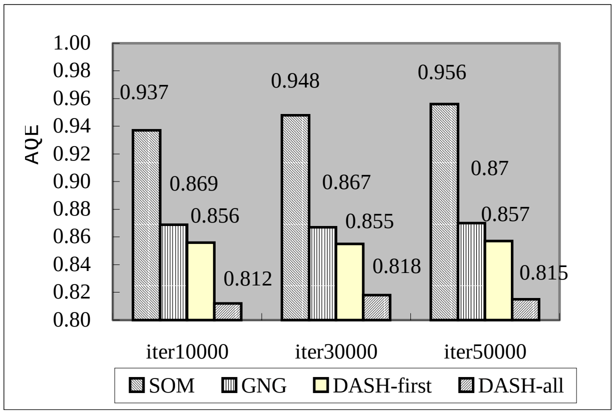
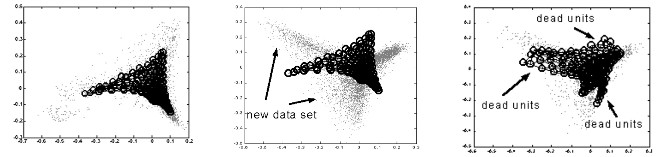


Fig. 5. A comparison of SOM, GNG and DASH evaluated by AQE. DASH-first denotes the first layer of the DASH hierarchy and DASH-all denotes the whole DASH hierarchy.

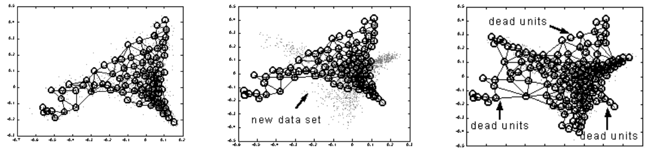
To demonstrate the convergence procedure step by step for three models, the first two principle components are used to project input samples and output units from a multi-dimensional space to a two-dimensional space. Each model uses three convergence maps to illustrate the behaviour when the new input set is explored at iteration 30,000 [Fig. 6-8]. For illustration, each of the 10,000 news articles is represented by a small grey dot, each unit is represented by a circle and each connection of units is represented by a line.

All models represent the existing data set well [Fig. 6a, 7a and 8a] and need to modify learned units to track the new shape of input samples due to the introduction of the new data set at iteration 30,000 [Fig. 6b, 7b and 8b]. However, SOM hardly adapts to the new data set because of the decaying learning rate and finally contains several dead units, which contain no associated input samples [Fig. 6c]. For GNG, a local connection-trimming procedure only removes unsuitable units connecting to recently activated units directly. In consequence, several unsuitable units which are too far away from recently activated units cannot be removed [Fig. 7c]. Unlike SOM and GNG, the DASH model removes unsuitable units and represents the new data set without dead units [Fig. 8c].



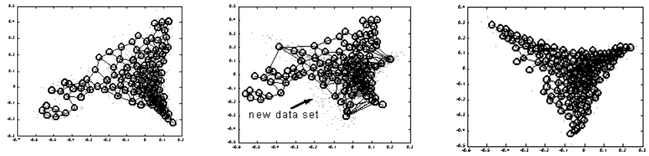
(a) iteration 29,000 (b) iteration 30,000 (c) iteration 46,000

Fig. 6. SOM after 29,000, 30,000 and 46,000 iterations in scenario 2. Each map contains 144 units and each of the 10,000 news articles is represented by a small grey dot. The first two principal components of 1,000-dimensional vectors are used as axes.



(a) iteration 29,000 (b) iteration 30,000 (c) iteration 46,000

Fig. 7. GNG after 29,000, 30,000 and 46,000 iterations in scenario 2. Each of the 10,000 news articles is represented by a small grey dot. The first two principal components of 1,000-dimensional vectors are used as axes.



(a) iteration 29,000 (b) iteration 30,000 (c) iteration 46,000

Fig. 8. DASH after 29,000, 30,000 and 46,000 iterations in scenario 2. Each of the 10,000 news articles is represented by a small grey dot. The first two principal components of 1,000-dimensional vectors are used as axes.

VII. CONCLUSION

In the non-stationary environment, a clustering model runs continuously since the new document set is formed consecutively for training while the old document set is still at the training stage. Thus, output units of the map learned from the old data set are continuously adjusted to reflect the new data set. Based on the same or very similar resources (i.e. training length and the number of units), the DASH model outperforms SOM and GNG in a non-stationary environment by a greater classification accuracy and a lower average quantization error. There are three main reasons as follows. Firstly, the DASH model uses fixed learning rates to keep the learning ability at any training stage so that the model can learn continuously. Secondly, the DASH model uses a global connection-trimming function to remove unsuitable units. This function helps the DASH model to represent input samples by its output units efficiently. Thirdly, the DASH model contains the self-adjusting connection-trimming threshold, which can be adapted by the old data set and be directly used for the new data set. This function helps the DASH model to use less training time than other models to achieve similar performance for the new data set.

REFERENCES

[1] J. Blackmore and R. Miikkulainen, “Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map,” *Proceedings of the IEEE International Conference on Neural Networks (ICNN’93)*, San Francisco, CA, USA, 1993, pp. 450-455.

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

[3] 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.

[4] 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.

[5] B. Fritzke, “Growing cell structures – a self-organizing network for unsupervised and supervised learning,” *Neural Networks*, vol. 7, no. 9, 1994, pp.1441-1460.

[6] F.H. Hamker and H.-M. Gross, “A lifelong learning approach for incremental neural networks,” *The Fourteenth European Meeting on Cybernatics and Systems Research (EMCSR'98)*, Vienna, 1998, pp. 599-604.

[7] 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.

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

[9] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero and A. Saarela, “Self organization of a massive document collection,” *IEEE Transactions on Neural Networks*, vol. 11, no. 3, 2000, pp. 574-585.

[10] R. Lang and K. Warwick, “The plastic self organising map,” *IEEE World Congress on Computational Intelligence*, 2002.

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

[12] T.M. Martinetz, “Competitive Hebbian learning rule forms perfectly topology preserving maps,” *Proceedings of ICANN-93, the International Conference on Artificial Neural Networks*, Amsterdam, 1993, pp. 427-434.

[13] K. McGarry and J. MacIntyre, “Knowledge Transfer between Neural Networks,” *Proceedings of the Sixteenth European Meeting on Cybernetics and Systems Research*, Vienna, Austria, April, 2002, pp. 555-560.

[14] A. Rauber, D. Merkl, and M. Dittenbach, “The growing hierarchical self-organizing maps: exploratory analysis of high-dimensional data,” *IEEE Transactions on Neural Networks*, vol. 13, no. 6, 2002, pp.1331-1341.

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

[16] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing & Management*, vol. 24, no. 5, 1988, pp. 513-523.

[17] S. Thrun and J. O'Sullivan, “Clustering learning tasks and the selective cross-task transfer of knowledge,” *Technical Report CMU-CS-95-209*, Carnegie Mellon University, School of Computer Science, Pittsburgh, 1995.