World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:4, 2014

Growing Self Organising Map Based Exploratory Analysis of Text Data

Sumith Matharage, Damminda Alahakoon

Abstract—Textual data plays an important role in the modern world. The possibilities of applying data mining techniques to uncover hidden information present in large volumes of text collections is immense. The Growing Self Organizing Map (GSOM) is a highly successful member of the Self Organising Map family and has been used as a clustering and visualisation tool across wide range of disciplines to discover hidden patterns present in the data. A comprehensive analysis of the GSOM's capabilities as a text clustering and visualisation tool has so far not been published. These functionalities, namely map visualisation capabilities, automatic cluster identification and hierarchical clustering capabilities are presented in this paper and are further demonstrated with experiments on a benchmark text corpus.

Keywords—Text Clustering, Growing Self Organizing Map, Automatic Cluster Identification, Hierarchical Clustering.

I. Introduction

THERE has been a massive increase in use of electronic documents in the recent past due to the proliferation of World Wide Web. At the same time, more sophisticated hardware technologies become more economical and feasible; therefore, organisations tend to store most of their data in digital format [1]. Also, as digital data provides a safer and more compact medium for data, almost everything is now stored in electronic format. Among these massive volumes of available data, textual data plays a vital role [2]. The increase in textual data has resulted in rich sources of data containing valuable and useful information for many applications across diverse disciplines such as social media and biomedical data analysis. This presents a challenge to maximise the use of this textual information effectively with minimum human intervention.

The field of text mining is emerged as an answer to this challenge. Text categorization is one major research area of text mining. Text categorisation is the process of grouping documents in a supervised manner based on the predefined labels. However this does not lead to discovery of any new information. Also, this is not applicable in many real world scenarios due to the unavailability of predefined labels. On the other hand, text clustering is emerged as a technique to automatically discover patterns present in text collections. This provides a way to view text data mining as a process of exploratory data analysis.

S. Matharage is with the School of Information and Business Analytics, Faculty of Business and Law, Deakin University, Victoria, 3125, Australia. (e-mail: s.matharage@deakin.edu.au).

D. Alahakoon is with the School of Information and Business Analytics, Faculty of Business and Law, Deakin University, Victoria, 3125, Australia. (e-mail: d.alahakoon@deakin.edu.au).

Out of several text clustering techniques proposed in the text mining literature Latent Dirichlet Allocation (LDA) based models, K-Means Clustering based models and Self organising Map (SOM) based models have shown great promise. All these facilitate finding hidden patterns in the text data. However, providing mechanisms to easily browse and navigate text collections are very important aspects of a text clustering system. The SOM [3], [4] is a neural network based clustering algorithm highly recognised for its visualisation capabilities. Therefore, it has been shown to be one of the best text clustering and visualisation algorithms [5]. After the initial success of the SOM in text clustering tasks, a family of SOM based algorithms has been developed. The Growing Self Organizing Map (GSOM) [6] is a highly successful member of the SOM family. It has shown great promise in many different data clustering tasks. Furthermore, capabilities of the GSOM have been extended and utilised across diverse disciplines [7], [8], [9], [10], [11]. Even though the GSOM has been widely used for different data clustering tasks there has been no comprehensive study of text analysis capabilities of the GSOM presented in the literature. In this paper we are addressing this gap highlighting the performance, automatic cluster separation and Spread Factor (SF) based hierarchical clustering capabilities of the GSOM when applied as a text clustering algorithm.

A brief overview of the GSOM algorithm, automatic cluster identification and hierarchical clustering capabilities of the GSOM are presented in the next section. The GSOM based text clustering model is presented in Section III. Section IV documents the experimental results and discussion in detail and Section V concludes the paper.

II. The Growing Self Organising Map

The Growing Self Organising Map (GSOM) addresses the issues associated with the predefined static architecture of the SOM algorithm. In the GSOM, the map of neurons is dynamically grown to reflect the topology of the input data set. Similar to most of the neural network based algorithms, the GSOM has two modes of activation, namely training and testing. The actual network growth and smoothing out of the weights occur during the training mode and final calibration of the network with known inputs takes place in the testing mode.

The training mode consists of three phases; namely, initialisation phase, growing phase and smoothing phase. A map with four neurons is initialised in the initialisation phase allowing the map to grow in any direction during the growing

phase. Input data is presented one after the other during the growing phase and the most similar neuron (winner) is identified using a general distance measure. After finding the winner weight vectors of the winner and its neighborhood are updated towards the current input. Also, the overall quantization error of the winner is updated and the new neurons are added to the map based on the accumulated quantisation error and the network boundary conditions of the winner. Finally, the error values of neurons are smoothed out in their neighborhood during the smoothing phase. Refer [6] for more detailed analysis of the GSOM algorithm.

Furthermore, the GSOM's ability of fitting itself into the input data distribution and spreading out according to the respective spread factor have been investigated in [12]. To demonstrate this, the resulting SOM and GSOM map structures for a star shaped 2D data set is illustrated in Fig. 1.

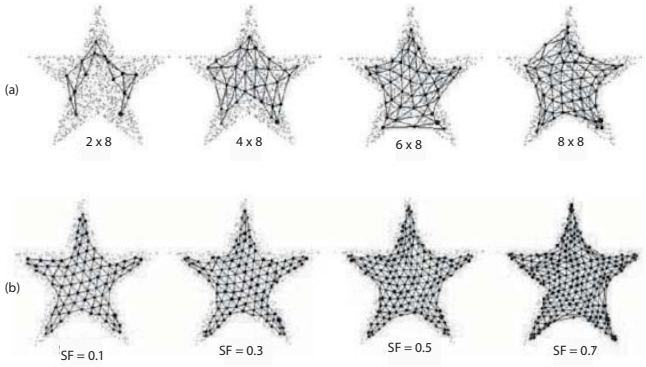

Fig. 1. (a) Different SOM structures mapped with Star data (b) The GSOM with different SF mapped with star data

It is very clear from Fig. 1 that the GSOM represents the data distribution more accurately than the SOM. It is clearly seen that the shape of the SOM has a major impact on how the network fits the input data. As the grid structure of the SOM becomes more and more square shaped, the network fits the input data more accurately compared to the rectangular networks since the star shape is closer in shape to a square. This will be a problem when the structure of the input data is unknown. However GSOMs with different spread factor (SF) values fits the same star shaped data for all four SF values. The effect of the SF is to control the level of detail of the network by increasing the number of neurons.

A. Automatic Cluster Identification in the GSOM

Automatic cluster identification is a very important task in the clustering process. Generally, clusters in a GSOM map can be identified by human inspection. In this approach, clusters can be identified as groups of hit nodes which are separated by dummy nodes or non-hit nodes. This is very feasible with higher values of the SF as it will generate a detailed map with increased branching out of the map. On the other hand, lower values of the spread factor will result in a low spread map, in which two groups of hit nodes might not be separated by non-hit nodes thereby forming unclear cluster boundaries. As shown in [13], it is essential to have an automatic cluster identification mechanism when the cluster boundaries are not clear.

The advantages of automatic cluster identification can be summarised as follows.

1) Reduces the ambiguity in cluster identification when the cluster boundaries are not sufficiently clear.
2) Faster Processing due to no human involvement.
3) Facilitates online processing and monitoring in a continuous processing environment.

Of the many different automatic cluster identification approaches, the K-Means and Davies-Bouldin (DB) index [14] based method and Data Skeleton Model (DSM) based method [15] are discussed in this section due to their promising results in identifying clusters in the GSOM maps. Even though these automatic cluster identification methods have been used across many application domains, suitability of these techniques for the text clustering domain has not been properly discussed. Therefore, we provide a detailed analysis of both techniques as cluster identification techniques particularly targeted at text clustering tasks. Two detailed algorithms are presented in Algorithm 1 and Algorithm 2, followed by experiments and discussion in section IV.

1) K-Means and DB Index Based Cluster Separation: The K-Means algorithm is a partitional clustering algorithm based on a minimum squared error criterion in grouping data. As proposed in [16], the usage of the simple K-Means algorithm together with the DB index as a cluster identification technique in a GSOM output map is presented in this section.

Algorithm 1: The K-Means and DB Index Based Cluster Separation Algorithm

input: N - number of hit nodes
for $k \leftarrow 1$ **to** $\sqrt{N}$ **do**
  Initialise k cluster centroids based on weight vectors of k randomly chosen hit nodes.
  repeat
    for *input* $x \leftarrow 1$ **to** N **do**
      Find the nearest centroid C, using a distance function,
      $C = \arg \min_{j \in N} \|x - w_j\|$
      Assign x to centroid C.
    Recalculate the new cluster centroids as,
    $C_i = \frac{\sum_{k=1}^{M} x_{k,i}}{M}$
    C_i - weight value at the i^{th} index of the centroid, M- number of assigned hit nodes, $x_{k,i}$ - weight value at the i^{th} index of the k^{th} hit node
  until *convergence of cluster centres or minimal or no change in the cluster membership*;
  Calculate the DB index,
  $DB = \frac{1}{n} \sum_{i=1, i \neq j}^{n} \max(\frac{S_i + S_j}{d(C_i, C_j)})$
  n - number of clusters, S_i, S_j - within cluster variations of cluster i and j, $d(C_i, C_j)$ - inter-cluster variation between cluster i and j
Find the k value minimises the value of the DB index.

In the K-Means algorithm the value of K needs to be pre-determined. This is a major limitation in the algorithm, especially when very limited or no knowledge about the data set is available. But as a general rule, 1 and $\sqrt{N}$ (N is the number of hit nodes present in the map) is used as the boundary values for K, when K-Means is used as a cluster separation technique. As shown in Algorithm 1, the K value which minimises the value of the DB-Index is selected as the final number of clusters. The applicability of this algorithm and choosing the best K-value when applied in text clustering tasks is discussed under Section IV.

2) Cluster Separation Based On Data Skeleton Model: Data Skeleton Modeling (DSM) has been proposed as an automatic identification of clusters in the GSOM [15]. The path of the spread (POS) is traced by linking the newly grown neurons to its parent neurons during the growing phase of the GSOM. After generating the data skeleton, the links corresponding to higher error values (higher inter node distances) are removed and the process is repeated until the required level of clusters is obtained. The final clusters can be selected dynamically by a data analyst by varying the separation threshold until the required level of clustering is achieved, or a predefined threshold value can be used to remove the links having an error value less than the threshold. The detailed algorithm is included in Algorithm 2.

Algorithm 2: Data Skeleton Model Based Cluster Separation Algorithm

Identify the path segments in the Path Of Spread (POS) in the Data Skeleton Model.
Calculate the distance D between all neighboring junctions using Euclidean distance,
$D_{x,y} = \sum_{i=1}^{Dim} (w_{i,x} - w_{i,y})^2$
x, y - two neighboring hit nodes, $D_{x,y}$ - Distance between x and y, w_i - weight value at the i^{th} index
repeat
  Calculate maximum distance between two neighbor nodes, $D_{max} = D_{x,y}$, such that $D_{x,y} \geq D_{i,j}$, $\forall_{i,j} \in N$ where N - set of hit nodes.
  Delete the segment xy.
until *satisfied with the cluster separation or meets the cluster separation threshold*;

B. Hierarchical Clustering with the GSOM

Hierarchical structure is a commonly used data structure in knowledge discovery and data mining [17]. It allows visualising the data at different levels of granularity and facilitates the identification of very detailed level relationships by drilling down to the abstract level relationships. Also, as psychological theories behind human learning highlight the importance of concepts at different granularity levels and the relationships between them, this provides insights into relating the clustering process to human learning and memory.

In the GSOM, the spread factor parameter provides a mechanism for controlling the growth of the network. In detail, a lower spread factor value generates a more abstract representation and that can be further explored by using a higher spread factor value. Therefore, for a given data set, a different set of clusters can be further explored by increasing the spread factor value after obtaining an initial abstract map. The process of hierarchical clustering is illustrated in Fig. 2.

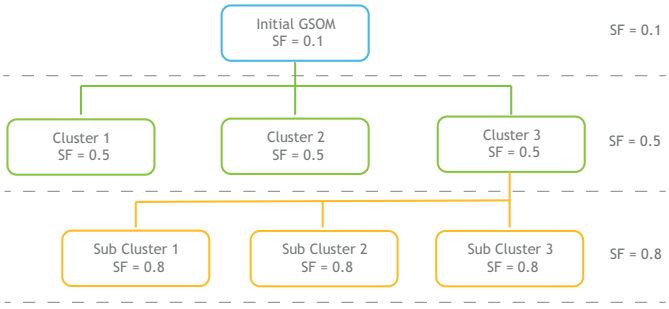

Fig. 2. Hierarchical clustering using the GSOM with multiple SF values

As shown in Fig. 2, an initial abstract map can be obtained using a lower SF value. Then the resulting clusters can be further explored using a higher SF value. This can be continued until the required number of levels is achieved. How these hierarchical clustering capabilities can be used in a text clustering task is explained in Section III.

III. The GSOM based Text Clustering Model

How the GSOM along with the above mentioned automatic cluster identification techniques and hierarchical clustering capabilities can be used in text analysis is discussed in this section.

Fig. 3 illustrates the GSOM based text clustering process. Initially, the input document collection needs to preprocessed in order to identify the features to be used in the GSOM clustering process. Generally, the individual words or sequences of words are extracted as the features for the clustering task. Stop words are removed and words are stemmed to their base form during the feature extraction. Also some thresholding will be applied to remove the very frequent and very infrequent features appear in the document collection as their contribution towards grouping document is not significant. After preprocessing, the documents will be represented based on the Vector Space Model (VSM) and will be fed in to the GSOM for grouping.

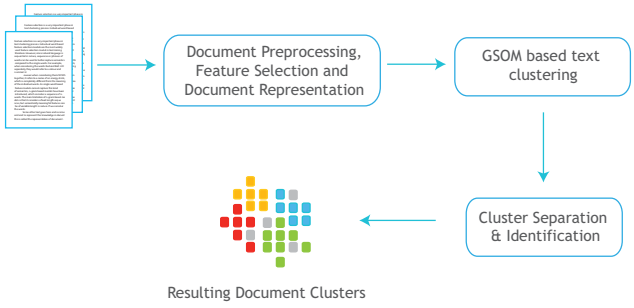

Fig. 3. The GSOM based text clustering model

The GSOM will automatically group the documents based on the document similarities from extracted features. Then the resultant GSOM map needs to be further processed to identify more meaningful document groups. Above mentioned K-Means and DB Index based approach or Skeleton Model based approach can be used to separate and identify the document clusters from the GSOM map. The applicability of both methods in text clustering tasks are further discussed along with the experiment results in Section III.

Previous research work on document clustering has shown that a collection of documents have an inherent natural hierarchical structure [18]. The flexible structure of the GSOM with its hierarchical cluster generation capability allows such inherent hierarchies to self represent themselves compared to the fixed structure SOM.

The number of features or attributes selected in text clustering tasks are dependent on the document collection. Generally, a representative feature set which covers the entire document collection without compromising the computational complexity of the clustering algorithm is selected. In the process of hierarchical text clustering, document collection selected for subsequent clustering can change when a lower level of the hierarchy is selected. As the feature set used in text clustering tasks is heavily dependent on the document collection, using the initial feature set is not meaningful at this stage for subsequent clustering. A new feature set that covers the new document subset must be selected instead. This would help to get rid of the unnecessary features (features present in the entire collection but not significant in the data set selected subsequently) and include the features that are not globally significant but are significant in the new subset of documents. Such dynamic feature selection mechanism has not been considered in most of the existing hierarchical text clustering; instead the same original feature set is used to arrive at clustering at different granularity levels. According to our knowledge, this is the first time that the hierarchical text clustering capabilities of the GSOM have been used in combination with a dynamic feature selection approach. This novel algorithm is presented in Algorithm 3 followed by the experiments in Section IV.

Algorithm 3: The GSOM Based Hierarchical Text Clustering Algorithm

```
Initialise SF to a lower value.
I = Input Collection
Cluster(I) function
repeat
    Select D = featureSet(I) based on thresholds, I - input collection.
    Generate weight matrix W using generalised Term Frequency as the term weighting technique.
    Feed W to GSOM G with a low SF value.
    Let's take C = Clusters(G).
    Increase the SF to next level.
    for cluster i ← 1 to N do
        N - number of clusters.
        I = DocumentSet(C_i) - assigns the document set mapped into the cluster C_i as the input collection Cluster(I). - this recursively calls the clustering function
until required level of spread is achieved;
```

IV. Experimental Results and Discussion

This section provides a set of experiments to demonstrate the capabilities of the GSOM as an exploratory text analysis tool. Reuters - 21578 distribution 1.0 data set is used as a benchmark data set to analyse these algorithms' capabilities in text clustering domain, since Reuters collection has been successfully used across a wide range of text mining tasks.

1) Description of Data Set and Preprocessing: The Reuters 21578, Distribution 1.0 data set [19] is a publicly available version of the well-known Reuters-21578 ApteMod corpus for text categorization. In the preprocessing stage, stop words were removed by maintaining a predefined list of stop words and words were stemmed to its base form using Porter's stemming algorithm. Also, appropriate lower and upper threshold values were applied during the feature selection process to remove the unwanted features that do not contribute significantly to the clustering. Normalised Term Frequency (NTF) was used as the term weighting technique. Generated document weight vectors based on Vector Space Model (VSM) were used as the input for the GSOM clustering.

2) Experiment 1: GSOM as a Text Clustering Tool: This experiment was carried out to identify the clusters present in a text corpus using the GSOM algorithm. As the resulting map for the entire Reuters collection was not easy to present and understand, a small document subset representing 5 major categories, namely *Acquisition*, *Earning*, *Interest*, *Trade* and *Crude* were chosen in this experiment. A total of 250 documents representing 50 documents from each category were selected. Preprocessed documents were fed into a GSOM algorithm with a 0.8 SF value. A 0.1 learning rate, 50 training iterations, 100 smoothing iterations, a 0.2 factor of distribution and a neighborhood radius of 3 were used as the other parameters. The resultant GSOM output map is presented in Fig. 4 (a).

Similarly, the same subset of the Reuters collection was clustered with a SOM for comparison. The SOM map size was based on the resulting GSOM map and found to be an 18×18 map. A 0.1 learning rate, 150 iterations and a neighborhood radius of 3 were used as the other parameters. The resulting SOM map is presented in Fig. 4 (b).

In the resulting map dark blue nodes correspond to the high density hit nodes and light blue nodes correspond to the low density hit nodes. Non-hit nodes were coloured in light grey. Clusters were identified by visual inspection using the knowledge about the categories present in the data set. Identified clusters were bounded using different colours and were labeled using the name of the most prominent category present in them.

When analysing the two maps, we can observe that the total number of neurons in the SOM map is significantly greater than that in the GSOM. This is because neurons are added

Fig. 4. (a) Resultant GSOM map (b) Resultant SOM map

only when necessary in the GSOM, but complete network of nodes need to be predefined in the SOM. This has also resulted in having a higher number of non-hit nodes in the final SOM map, degrading the computational efficiency of the SOM algorithm compared to the GSOM.

This experiment shows how the GSOM can be utilised as a text clustering and visualising tool. A detailed analysis and comparison of other important aspects of the GSOM are demonstrated in the following experiments to highlight the significance of the GSOM algorithm for text clustering.

3) Experiment 2: Analysis of Cluster Separation in the GSOM: This experiment was conducted to demonstrate the functionality of the two cluster separation algorithms presented in Algorithm 1 and Algorithm 2.

Firstly, cluster separation based on the K-Means and DB-Index technique is presented. A resultant GSOM map from Experiment 1 was used as the input for cluster separation. The best K value which minimises the DB-index value was chosen as given in Algorithm 1 and it was two. The resultant map with two identified clusters is illustrated in Fig. 5 (a).

In the resulting map, the *Earning* cluster is clearly separated from the rest. All the other categories are grouped as a single cluster. This cannot be accepted as the appropriate level of granularity of interest, as this is significantly different from the clusters obtained by visual inspection. Therefore, we tried a trial and error approach to come up with the best K separation. K = 5 and K = 6 resulted in a separation quite close to the separation obtained by visual inspection. The resulting maps corresponding to K = 5 and K = 6 are presented in Fig. 5.

In the process of identifying the best K value using a trial and error based method, we started with K = 5, as visual inspection resulted in 5 clusters. The *Earning* cluster is clearly separated from the rest similar to that of K = 2. Documents belonging to the category *Crude* are distributed together with the *Acquisition* and *Trade* categories and documents belonging to the category *Interest* are split into two clusters. Therefore, K = 5 seems to provide a more meaningful separation than K = 2. We further analysed the resultant map with K = 6 and observed that the *Crude* and *Acquisition* cluster in the K = 5 map is split into 2 new clusters while keeping the other clusters the same. By considering all the facts, K = 6 was chosen as the best cluster separation for this clustering task.

Next, a Data Skeleton Model (DSM) based cluster separation method was used to identify the clusters in the same resultant GSOM map. A parameter called cluster separation threshold (δ), where $0 < \delta < 1$, is used to identify the resultant clusters. $\delta = 1$ corresponds to a very abstract map with one cluster and $\delta = 0$ corresponds to a map where each hit node is considered as a separate cluster. By changing the value of δ, the required level of separation can be achieved. When the value of δ was decreased gradually, more clusters appeared in the map. Three resultant maps for $\delta = 0.8$, $\delta = 0.6$ and $\delta = 0.3$ are presented in Fig. 6.

When $\delta = 0.8$, *Trade* and a part of *Interest* documents were separated from the rest as a cluster. When the value of δ was further reduced to 0.6, documents belonging to the category *Acquisition* appeared as a new cluster. Similarly, when $\delta = 0.3$ most of the clusters were clearly separated exhibiting a similar separation to human identified clusters. More finer grain clusters can be obtained by gradually decreasing the value of δ, but for this clustering task $\delta = 0.3$ was selected as the best separation.

In summary, we can conclude that both the techniques can be effectively used to identify the clusters in a GSOM map when applied in text clustering tasks. A limitation of the K-means and DB-Index based method is that the user might have to fine-tune the value of the best K, as minimising the DB-Index value might not always guarantee the best K for text clustering tasks. Further experiments showed that the best K value selected by minimising the DB-Index is more suitable when the map size is relatively small, but not when the map contains a large number of neurons. When the map is large K = 2 was found to be the best K. Therefore, a trial and error method has to be employed in those scenarios to identify the best separation. This might be computationally inefficient when there is no prior knowledge about the data set and also

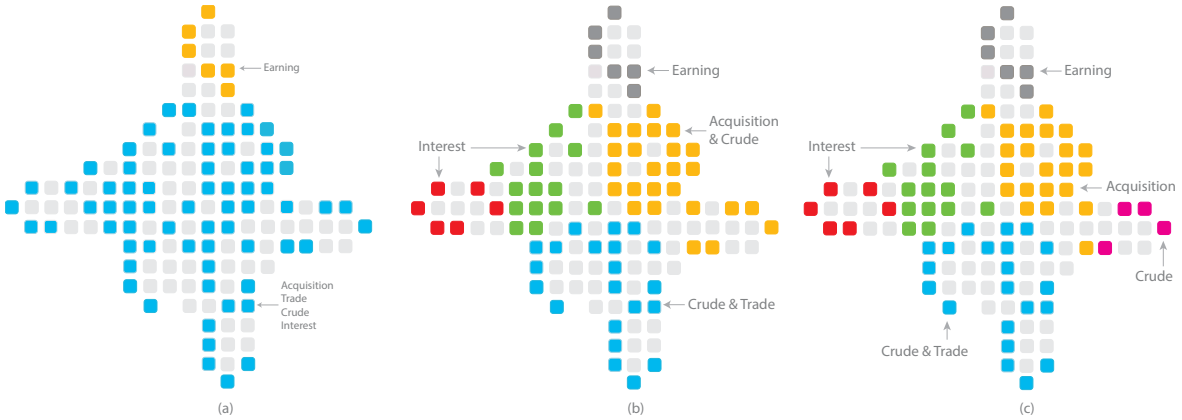

Fig. 5. K-Means based cluster separation (a) K = 2, (b) K = 5, (c) K = 6

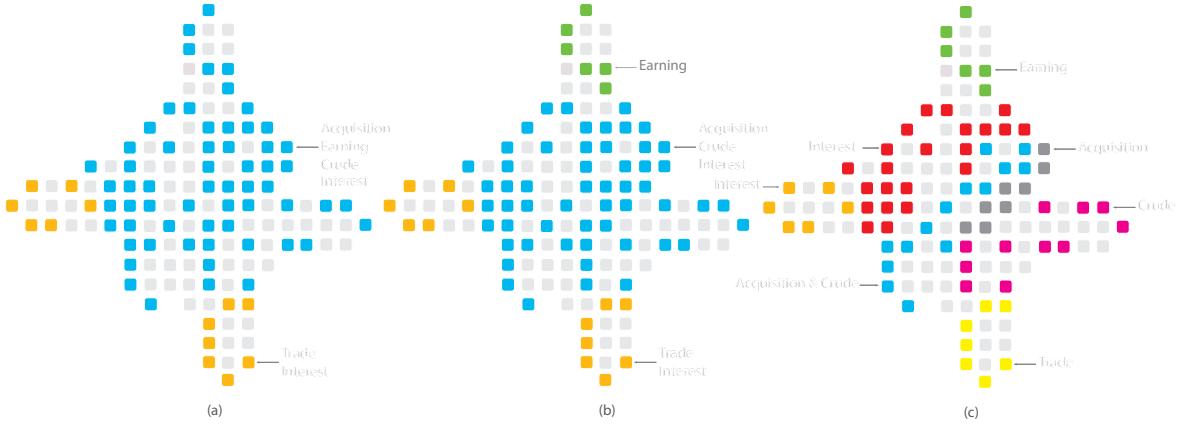

Fig. 6. Skeleton model based cluster separation (a) $\delta = 0.8$, (b) $\delta = 0.6$, (c) $\delta = 0.3$

when the map size is large. On the other hand, a skeleton model based method provides a mechanism to identify clusters at different granularity levels by using the parameter δ ($0 < \delta < 1$). This method is quite simple and easy to use compared to the K-Means and DB-Index based approach. But, in general, both methods can be used with properly selected parameter values to separate and identify document clusters in a GSOM map without human intervention.

4) Experiment 3: Accuracy and Efficiency Comparison with the SOM: This experiment was carried out to compare the GSOM text clustering capabilities to that of the SOM algorithm. Six different subsets of Reuters data set with 100, 250, 500, 1250, 2500 and 5000 documents were selected and clustered with a GSOM and SOM separately. SOM map sizes were selected based on the corresponding GSOM map size.

Lower document threshold (LDT) and upper document threshold (UDT) values were applied during the feature selection to remove the unwanted features which do not significantly contribute in the clustering process. Generally, features that appear very often, that is, in the majority of documents and those that appear seldom, that is, in few documents do not help in identifying distinguish clusters.

A formal comparison of computational times for the SOM and the GSOM was conducted by using the same subsets of the Reuters collection. A 0.1 factor of distribution, 50 training iterations and 100 smoothing iterations were used during the GSOM training and 150 iterations were used for the SOM training. A 0.1 learning rate and a neighborhood radius of 3 were used as the other parameters in both the algorithms. The average execution times over five executions were recorded for each clustering task under identical conditions in a computer with an Intel(R) Core(TM) i5-2400, a 3.10 GHz processor and 8GB of memory. Results are presented in Fig. 7.

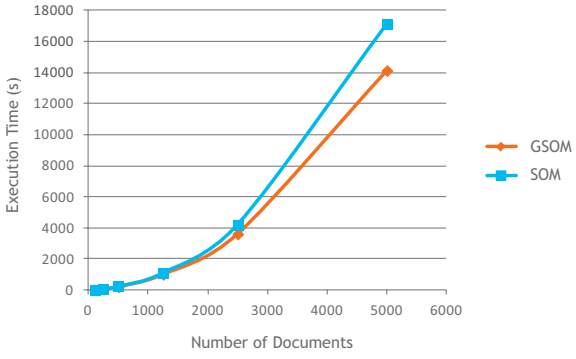

Fig. 7. Comparison of the execution times for the SOM and the GSOM

It can be clearly seen that the time taken by both algorithms increases with the number of documents in the data set. But

the GSOM outperforms the SOM in all the cases resulting better execution times. This is due to the small map size at the initial phase of the GSOM training. Also, it is important to highlight that unless the GSOM size was used as a basis to determine the SOM map size, trial and error attempts to decide the SOM map size would have further increased its computational time.

To compare the accuracy of the results, Precision, Recall and F-Measure values were used. These measures have been widely used in information retrieval literature [20], [21]. Table I summarises the Precision, Recall and F-Measure values of all 5 categories in the document subset with 1250 documents (250 documents from each category). Each experiment was carried out 5 times and the average values of the Precision, Recall and F-Measure were recorded. The results demonstrate that the GSOM preserves similar or better accuracy level in all 5 categories.

Based on all the experimental results, we can conclude that the GSOM provides a more efficient and accurate text clustering algorithm compared to the SOM.

5) Experiment 4: Hierarchical Clustering with the GSOM: In this experiment, the GSOM was used to explore the hierarchical relationships present in a text corpus. Initially, the data set was clustered with a 0.1 SF value and the resultant clusters were identified using the K-Means and DB-Index based cluster separation technique. This resulted in 3 clusters. Then the documents belonging to these 3 clusters were clustered separately with a 0.5 SF value. Feature sets were identified separately for each sub-clustering task. Third level clustering was obtained in a similar fashion using a 0.8 SF value. Part of the resulting hierarchical structure is presented in Fig. 8.

In the hierarchy obtained, more abstract level clusters can be observed at the top levels of the hierarchy, with more finely grained clusters at the bottom levels of the hierarchy. Significant terms representing the clusters were identified and included along with the cluster labels to provide a more meaningful representation of results. The terms identified at the bottom levels of the hierarchy were very specific and more significant than the terms identified at the top levels of the hierarchy.

V. Conclusion

The capabilities of the GSOM algorithm especially as a text clustering tool are evident from the above experiments. The GSOM provides efficient, more accurate and dynamic clustering capabilities compared to the SOM. Also, the resultant clusters can be visualised as a map of text clusters and those can be further explored to discover the hierarchical relationships present in the data. To our knowledge this is the first comprehensive analysis of text clustering capabilities of the GSOM that highlights the cluster separation and hierarchical text clustering capabilities as well as highlights the advantages over the traditional SOM.

In summary, key features and advantages of the GSOM algorithm as a text clustering and learning algorithm are enumerated below.

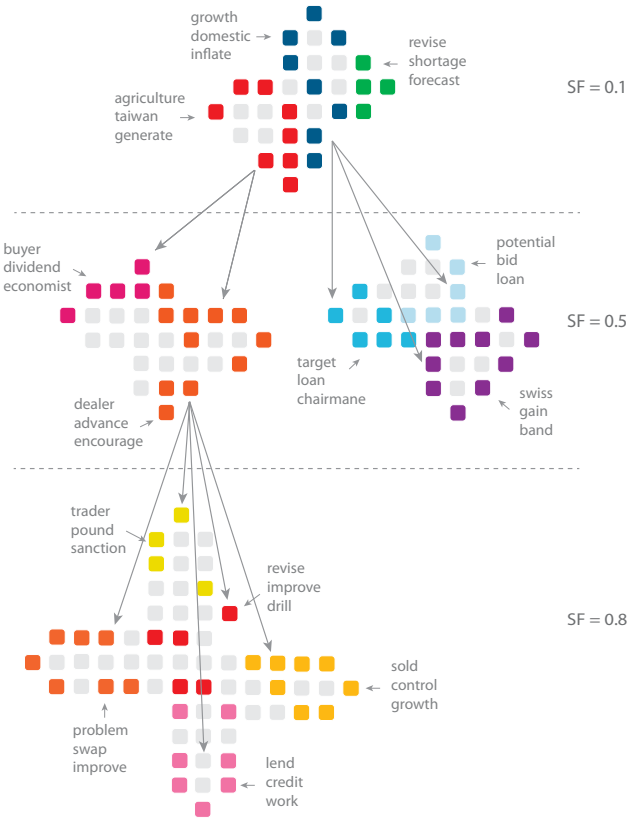

Fig. 8. The GSOM based hierarchical text clustering

1) *Structurally unconstrained learning* - the GSOM ensures that the learning outcomes are not restricted to a predefined architecture. It will generate the map structure based on the natural topology of the input document collection during the training of the algorithm.
2) *Exploratory dynamics* - the GSOM introduces a parameter named Spread Factor (SF) which controls the growth of the network. Smaller values of the SF will provide a more abstract view of the input data highlighting the key features. Higher values of the SF can be used when further explorations are required. This capacity for exploratory dynamics provides a way of discovering hierarchical structures present in document collections.
3) *Efficient computation* - the GSOM provides improved computational efficiencies compared to the SOM algorithm as it starts with a small map. Neurons will be added to the map when required.
4) *Visualisation* - the GSOM inherits visualisation capabilities of the SOM and therefore provides mechanisms to visualise the clusters of input text collections. Due to the cluster dictated map shape, the GSOM can result in better visualisation.

TABLE I
COMPARISON OF PRECISION, RECALL AND F-MEASURE FOR THE SOM AND THE GSOM

Category name	SOM			GSOM		
	Precision	Recall	F-Meaure	Precision	Recall	F-Measure
Earning	0.82	0.84	0.83	0.83	0.84	0.83
Acquisition	0.75	0.80	0.77	0.78	0.81	0.79
Trade	0.78	0.82	0.80	0.82	0.84	0.83
Crude	0.72	0.73	0.72	0.79	0.78	0.78
Interest	0.76	0.79	0.77	0.78	0.83	0.80

REFERENCES

[1] A. Haug, "The implementation of enterprise content management systems in smes," *Journal of Enterprise Information Management*, vol. 25, no. 4, pp. 349–372, 2012.

[2] D. Robb, "Text mining tools take on unstructured data," *Computerworld*, 2004.

[3] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[4] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, pp. 52–65, 2013.

[5] D. Isa, V. Kallimani, and L. Lee, "Using the self organizing map for clustering of text documents," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9584–9591, 2009.

[6] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan, "Dynamic self-organizing maps with controlled growth for knowledge discovery," *IEEE-NN*, vol. 11, no. 3, p. 601, May 2000.

[7] M. Cao, A. Li, Q. Fang, E. Kaufmann, and B. J. Kroeger, "Interconnected growing self-organizing maps for auditory and semantic acquisition modeling," *Frontiers in psychology*, vol. 5, 2014.

[8] C. D. Wijetunge, Z. Li, I. Saeed, J. Bowne, A. L. Hsu, U. Roessner, A. Bacic, and S. K. Halgamuge, "Exploratory analysis of high-throughput metabolomic data," *Metabolomics*, vol. 9, no. 6, pp. 1311–1320, 2013.

[9] K. Wickramasinghe, D. Alahakoon, P. Schattner, and M. Georgeff, "Self-organizing maps for translating health care knowledge: A case study in diabetes management," in *AI 2011: Advances in Artificial Intelligence*. Springer, 2011, pp. 162–171.

[10] P. Lokuge and D. Alahakoon, "Improving the adaptability in automated vessel scheduling in container ports using intelligent software agents," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1985–2015, 2007.

[11] S. Matharage, O. Alahakoon, D. Alahakoon, S. Kapurubandara, R. Nayyar, M. Mukherji, U. Jagadish, S. Yim, and I. Alahakoon, "Analysing stillbirth data using dynamic self organizing maps," in *DEXA Workshops*, F. Morvan, A. M. Tjoa, and R. Wagner, Eds. IEEE Computer Society, 2011, pp. 86–90.

[12] D. Alahakoon, "Controlling the spread of dynamic self-organising maps," *Neural Computing and Applications*, vol. 13, no. 2, pp. 168–174, 2004.

[13] R. Amarasiri, L. Wickramasinghe, and D. Alahakoon, "Enhanced cluster visualization using the data skeleton model," *Proceedings of Soft computing and the Web (ISCW)*, vol. 3, pp. 239–548, 2003.

[14] D. Davies and D. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 224–227, 1979.

[15] D. Alahakoon, S. Halgamuge, and B. Srinivasan, "Mining a growing feature map by data skeleton modelling," *Studies in fuzziness and soft computing*, vol. 68, pp. 217–250, 2001.

[16] N. Ahmad, D. Alahakoon, and R. Chau, "Cluster identification and separation in the growing self-organizing map: application in protein sequence classification," *Neural Computing and Applications*, vol. 19, no. 4, pp. 531–542, 2010.

[17] M. Schkolnick, "Clustering algorithm for hierarchical structures," *ACM Trans. on Database Sys.*, vol. 2, no. 1, p. 27, Mar. 1977.

[18] D. Merkl, "Text classification with self-organizing maps: Some lessons learned," *Neurocomputing*, vol. 21, no. 1-3, pp. 61–77, 1998.

[19] D. D. Lewis, "Test Collections : Reuters-21578," http://www.daviddlewis.com/resources/testcollections/reuters21578/, 2004, [Online; accessed 01-August-2009].

[20] C. Manning, P. Raghavan, and H. Schutze, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 1.

[21] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proceedings of DARPA Broadcast News Workshop*, 1999, pp. 249–252.

Sumith Matharage was born in 1983. He received his B.Sc Eng. (Honours) in Computer Science and Engineering from the University of Moratuwa, Sri lanka in 2007 and Ph.D. in Computer Science from Monash Unviersity, Australia in 2012.

He is currently working as an associate lecturer at School of Information and Business Analytics, Faculty of Business and Law, Deakin University, Australia where he is a member of the Deakin Cognitive Analytics Research Lab and SAS Visual Analytics Collaboratory at Deakin University. His current research interest includes unsupervised learning, self-organisation, visual analytics, predictive analytics and text mining. He has also worked in the IT industry as a Software Engineer and a Business Intelligence Analyst, in Sri Lanka and Australia.

Damminda Alahakoon Damminda Alahakoon is an Associate Professor at Deakin University, Australia, where he leads the Deakin Cognitive Analytics Research Lab as well as teaching in to subjects in Predictive Analytics and Business Intelligence. He received his BSc (Hons) in Computer Science from the University of Colombo, Sri Lanka in 1995 and his PhD degree from Monash University, Australia in 2002. Prior to his current appointment, he was an academic staff member at Monash University, Australia for 10 years. His research expertise lies in the areas of, data mining, text analytics, artificial intelligence and business intelligence. He has a special interest in self organising systems, self organising maps especially in adapting and evolving structures. Prior to his academic career, Damminda has worked in the IT and finance industries, in Sri Lanka, Australia and the Netherlands.