Web Content Management by Self-Organization

Richard T. Freeman, *Student Member, IEEE,* and Hujun Yin, *Senior Member, IEEE*

***Abstract*—We present a new method for content management and knowledge discovery using a topology-preserving neural network. The method, termed topological organization of content (TOC), can generate a taxonomy of topics from a set of unannotated, unstructured documents. The TOC consists of a hierarchy of self-organizing growing chains (GCs), each of which can develop independently in terms of size and topics. The dynamic development process is validated continuously using a proposed entropy-based Bayesian information criterion (BIC). Each chain meeting the criterion spans child chains, with reduced vocabularies and increased specializations. This results in a topological tree hierarchy, which can be browsed like a table of contents directory or web portal. A brief review is given on existing methods for document clustering and organization, and clustering validation measures. The proposed approach has been tested and compared with several existing methods on real world web page datasets. The results have clearly demonstrated the advantages and efficiency in content organization of the proposed method in terms of computational cost and representation. The TOC can be easily adapted for large-scale applications. The topology provides a unique, additional feature for retrieving related topics and confining the search space.**

***Index Terms*—Bayesian information criterion (BIC), content management, document categorization, hierarchical clustering, information retrieval (IR), self-organizing maps (SOMs), taxonomy generation, topic hierarchy, topological tree structure.**

I. Introduction

With the vast and increasing amount of unstructured content available electronically on the web and enterprise intranets it is becoming more and more inefficient to rely on humans to manually annotate electronic documents, e.g., webmasters and knowledge managers dealing with content categorization, updating and maintaining the file stores. This is why web content management has become an important area of research and application. The main applications are e-libraries, enterprise portals, e-commerce, software contents management, document management and knowledge discovery [1]. Document management addresses many issues such as storage, indexing, security, revision control, retrieval and organization. The documents, generated in an enterprise either centrally or locally by employees, are often unstructured or arranged in ad hoc manner (e.g., e-mails, reports, web pages, presentations). As employees are highly mobile or distributed with possibly varying locations, it is important to extract their contributions and knowledge and to systematically organize them into a knowledge portal. Recently many companies have recognized the value of content management and have focused on text mining and dynamic document management tools, for decision-making tasks or knowledge base creation [2]. Many existing full-text search engines return a large ranked list of documents, many of which are irrelevant. This is especially true when queries are short and very general words are used. Hence, the document organization has become important in information retrieval (IR) and content management. It includes the following.

- *Topic detection and tracking*: Use a stream of documents to detect topics that are tracked over time. Theses topics can be described as events, facts or activities. In topic detection and tracking, there are three distinctive steps, segmentation of a document into disjoint story boundaries, detection of new topics, and tracking of known topics (or topics of interest). The application has been particularly useful for tracking important news events [3].
- *Document classification*: Use a set of documents that are preclassified to specific topics, to create a model for each topic. New documents are then classified and assigned to existing topics with the most similar models. The typical supervised learning methods are the backpropagation neural networks [4], naïve Bayesian [5] or support vector machine [6]. The applications include e-mail filtering and categorizing documents. An introduction and review can be found in [7].
- *Document clustering*: Use a set of unclassified documents to extract important relations among them and organize similar ones together to form topic groups. Documents in a topic group share certain common properties. Typical methods include partition-based clustering [8], agglomerative/divisive hierarchical clustering [9], and self-organizing maps (SOMs) [10]. Document clustering has been applied to areas such as expanding search space [9], browsing large sets of documents [8] and arranging results from a web search engine query [11]. An extensive and detailed review can be found in [12].

The proposed topological organization of content (TOC) method provides a natural way to automatically generate topic directories and structures for efficient information access and retrieval and for document navigation and exploration—an important aspect in modern web content management [1] and document management [13] systems. The dynamically generated topic structures by the TOC method, the topological trees, share similarities with many major web directories such as Yahoo[1] or Dmoz,[2] however, are created directly and autonomously from a set of unlabeled documents. Instead, Dmoz, for example, requires about 60 000 human indexers to create, update and maintain the structure.

Manuscript received October 31, 2003; revised March 31, 2005. This work was supported by Engineering and Physical Sciences Research Council (EPSRC) U.K. studentship.

The authors are with the School of Electrical and Electronic Engineering, University of Manchester, Manchester M60 1QD, U.K.

Digital Object Identifier 10.1109/TNN.2005.853415

[1] http://www.yahoo.com

[2] http://www.dmoz.com

The main strength of the proposed TOC method is that the size of each chain is adaptive and self-determined through a validation process. This is novel compared to many existing methods that use predetermined and fixed growing threshold, effectively imposing constraints on the structure. The use of a hierarchy of one-dimensional (1-D) chains instead of grids matches more closely the user-friendly tree structures used by many file explorers, portals or web directories.

The remaining of the paper is organized as follows. A brief review on general document clustering methods as well as their respective validations is given in the next section. In Section III, the proposed TOC method is described. Its key components, the growing chains (GCs), extract the essence of the contents and use a proposed entropy-based Bayesian information criterion (BIC) to validate their structures. The TOC method forms a hierarchical organization of the contents in a topological tree structure for visualization. Section IV presents comparative results, demonstrating the benefits and advantages of incorporating dynamic growth and validation procedures for retrieval and exploration. Section V concludes the paper by discussing the potentials of the proposed TOC method compared with the existing techniques and suggests future work.

II. Existing Document Organization Methods

A. Document Clustering Algorithms

The main algorithms for document clustering are partitioning, hierarchical, information theoretic, probabilistic model-based, tree-based clustering and neural network based. Partition-based methods generally use a centroid to represent a cluster, and assign documents to clusters sequentially or in a batch. Examples include the sequential single-pass [14], batch k-means, fractionation and buckshot [8]. In hierarchical clustering, there are two types of algorithms, the divisive and agglomerative. The agglomerative algorithm begins with a single document as a cluster and iteratively merges the most similar clusters until a terminating criterion is met or all documents fall into a single cluster. Typical agglomerative algorithms include average-link [15] and Ward's method [16]. In the divisive clustering, all documents are assigned to a single cluster at the beginning and iteratively selected clusters are divided into smaller clusters. Examples are bisecting k-means [17], k-means variants [18], and principle direction divisive partitioning [19]. Detailed reviews of partition-based and hierarchical document clustering can be found in [12]. Information theoretic based methods use statistical methodology instead of distance or similarity metrics and can maximize the mutual information (MI) between documents and categories [20]. Probabilistic model-based methods use models, e.g., Gaussian mixture [21] or assume a probabilistic word distribution to represent clusters or documents. Tree-based clustering methods employ suffix tree clustering to represent and cluster the documents [11] or machine learning type decision trees [22].

There are few nonneural network methods that can generate browsable topic hierarchies. Most methods generate dendrograms used for retrieval. However, some methods create an n-way tree fixed at each level using k-means variants [8], [18]. A recent approach uses fuzzy relations to construct a hierarchy [23], however the structure is a directed acyclic graph that cannot be easily represented as a tree structure. Both methods do not have a topology making it difficult to visualize the similarities between the clusters. A thorough, complete review of other hierarchical methods can be found in [12].

B. Neural Networks for Document Organization

The benefits of neural networks such as nonlinear input–output mapping and ability to learn from examples have made them suitable for document organization. The most widely used neural networks in document organization are the adaptive resonance theory (ART) and SOMs. The ART is a method guided by a vigilance parameter, which helps to determine if the network represents a chosen sample sufficiently well. It has been used for clustering documents, e.g., the basic ART in [24] or Fuzzy ART [25]. Kohonen's SOM and its variants [26], allow clustering and visualization of documents and their relationships in a two-dimensional (2-D) representation. A typical example is the WEBSOM [10]. Many variants of the SOM have been proposed and applied to document organization, e.g., Marginal Median SOM [27], TreeGCS [28], and growing hierarchical-SOM (GH-SOM) [29]. The main advantage of the SOM is the topology preservation of input space, which makes similar topics appear closely on the map. The disadvantages include the complexity that increases with the number of units and size of the dataset, inability to handle efficiently dynamic document sets, unlike the plasticity in the ART for example. More detailed discussions on existing SOM-based algorithms applied to document organization can be found in [12].

An important part of visualization and exploration of the SOM-based methods is to automatically identify and label regions of interest. Multilayered SOMs have been used to organize documents relating to entertainment and brainstorming (e.g., the ET-MAP [30]), where each node is labeled using the best matching terms and nodes with similar terms are merged together to form regions on the maps. This is similar to the approach adopted in the LABELSOM [29], which uses the quantization error (QE) to determine the best terms to label the nodes without merging nodes into regions. The WEBSOM uses a term frequency-based measure to determine the labels [10]. The method looks at frequencies of the most discriminating terms occurring in a cluster. The WEBSOM uses more nodes than the other methods, so not every node is labeled, but rather there is a fixed radius of minimum distance between labels. On the map the cluster densities are colored and smoothed, making them more visually appealing than the ET-MAP and GH-SOM. The WEBSOM also reduces the term space using the random projection to create the document vectors.

C. Quantitative Validation of Document Clustering

This section focuses on document-clustering validation metrics that do not require hand sorted document sets. These unsupervised learning methods are more appropriate as objective solutions, when subjective solutions are not available, especially in huge, distributed information resources. Supervised clustering validation can use for example generic cluster validation methods [31] and supervised evaluation methods used in IR such as precision-recall [9].

A nonexhaustive list of document validation metrics is presented in the following.

- The overall similarity (or dissimilarity) criterion between each document and its nearest cluster representative for all n clusters
$$\text{IntraSimTotal} = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{m_k}\sum_{\mathbf{x}\in C_k}\text{Metric}(\mathbf{x},\mathbf{c}_k) \quad (1)$$
where $\mathbf{x}$ is a document vector, $\mathbf{c}_k$ is the mean of cluster C_k, m_k is the number of items in C_k and Metric is the similarity (e.g., cosine correlation) or distance (e.g., Euclidean) measure used. This measure has been typically used in k-means training and for the mean-squared-error validation where the turning point may indicate the desired number of clusters.
- In the intracluster similarity or pairwise similarity of each document in the same group
$$\text{IntraSimPair}(C_k) = \frac{1}{m_k(m_k-1)}\sum_{\substack{\mathbf{x},\mathbf{y}\in C_k\\ \mathbf{x}\neq\mathbf{y}}}\text{Metric}(\mathbf{xy}) \quad (2)$$
where $\mathbf{x}$ and $\mathbf{y}$ are documents clustered to C_k. This measure is typically used to calculate the entries in a similarity matrix in hierarchical agglomerative methods.
- The cluster cohesion measure looks at the common words within a cluster
$$\text{cohesion} = \sum_{k=1}^{n}\frac{m_k}{m}c(t\,|\,C_k) \quad (3)$$
where m is the total number of documents in collection and $c(t\,|\,C_k)$ the count of terms in C_k. This measure was applied in [32] and [33].
- In a model based method, the MI is used to find a stable model over many runs [21]. The MI is used to measure the similarity between two sets of clusters
$$\text{MI}(S_1,S_2) = \sum_{b_i\in S_1, c_j\in S_2} p(b_i,c_j)\cdot\log_2\frac{p(b_i,c_j)}{p(b_i)\cdot p(c_j)} \quad (4)$$
where $S_1 = \{b_1,\ldots,b_n\}$ and $S_2 = \{c_1,\ldots,c_n\}$ are sets of documents clusters, $p(b_i,c_j)$ is the joint probability that a document belongs to both S_1 and $S_2, p(b_i)$, and $p(c_j)$ are the probabilities that a randomly selected document belong to cluster S_1 or S_2, respectively.
- Entropy: this is a measure that can indicate the coherence of a cluster. The entropy has been used to quantitatively compare algorithms in [17] and [19]. The entropy $H(C)$ of cluster C can be defined as
$$H(C_k) = -\sum_i p(i\,|\,C_k)\log p(i\,|\,C_k) \quad (5)$$
where $p(i\,|\,C_k)$ is the probability that a term i belongs to cluster C_k. The entropy is an efficient validation method and does not require a model when the probabilities are obtainable. It constitutes the basis of the validation method used in this paper (see Section III-F).
- The BIC [34] can also been adopted to validate document clustering in a model-based approach. The optimum number of models can be found using
$$M_{\text{BIC}} = \arg\max\left\{M_j(X_1,X_2,\ldots,X_q) - \frac{1}{2}y_j\log q\right\} \quad (6)$$
where M_j is the model $j(j=1,2,\ldots,q), X$ an associated sample, y_j the number of dimensions in the model j and q the sample size.

D. Validation and Evaluation of SOMs

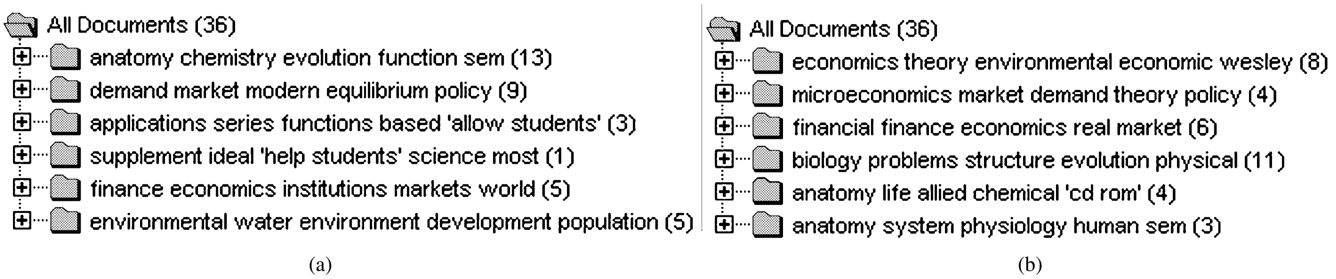
The validation of the SOM is still an unexplored area and even more so for document organization. The focus has been given to proofs of convergence [35], weight ordering [26], and reliability measures [36]. Davies–Bouldin index and a combination of validation indexes have been used in the SOM for clustering a nondocument dataset and compared with other clustering algorithms [37], [38].

Little work has been done to validate document maps themselves. If the SOM is used to perform clustering then any existing validation methods from cluster analysis can be used. However, these methods do not take into account the topology preservation property. The ART, SOM, as well as agglomerative clustering, AUTOCLASS, generative topographic mapping, SOM and GH-SOM, have been used to organize documents and the results compared empirically [39]. Supervised measures of precision-recall have been used but require known class memberships of documents prior to being applied to clustering. In such a comparison, better results were achieved than those of existing full-text search engines [40]. One method evaluating the Marginal SOM and basic SOM found that in terms of precision and recall the former provided better results [27]. However, precision-recall has been regarded as unsuitable for evaluating the SOMs, as it assumes a optimum solution already exists [41]. Instead, the various measures such as map usage, average intranode similarity, fragmentation (number of isolated map regions) and purity (intracluster distance) have been used. However, the validations were made on the indexing terms but not the maps or clusters [41].

III. Proposed TOC Method

A. Importance of Topology and Tree View Organization

Generally most clustering algorithms ignore the overlying relationships between clusters, as they are typically concerned with retrieval of documents rather than browsing. For example agglomerative methods generate dendrograms that are difficult to navigate if there are a large number of documents. Although they look at the similarity between documents, they do not clearly mark, represent and visualize the similarities among all clusters. Likewise flat partitioning algorithms such as k-means do not show the similarities between cluster representatives. However, it is advantageous for browsing, to obtain and visualize similar topics close to one another even in 1-D structures as it can be observed in Fig. 1.



All Documents (36)
- anatomy chemistry evolution function sem (13)
- demand market modern equilibrium policy (9)
- applications series functions based 'allow students' (3)
- supplement ideal 'help students' science most (1)
- finance economics institutions markets world (5)
- environmental water environment development population (5)

(a)

All Documents (36)
- economics theory environmental economic wesley (8)
- microeconomics market demand theory policy (4)
- financial finance economics real market (6)
- biology problems structure evolution physical (11)
- anatomy life allied chemical 'cd rom' (4)
- anatomy system physiology human sem (3)

(b)

Fig. 1. k-mean with (a) no topology and (b) the proposed 1-D Growing SOM (GCs).

The 1-D self-organizing networks can produce a useful topology, which the k-means method is incapable of. An example is shown in Fig. 1, where in the k-means the topics are not ordered at all, while in the SOM-based result the neighboring topics are closely related. Topology is important for browsing and organizing contents and retrieving relevant information, as it produces better structures and builds relationships among the documents, especially since the labels are assigned automatically when alphabetical ordering may not be suitable. This allows users to directly infer the general topics and their relations. The k-means with random initialization lead to different solutions over many runs. This is a widely known limitation of the algorithm, e.g., [42]. Whereas topological mappings have been found to be more consistent over many runs and can cope well with random initializations.

Tree view representation of content is a natural way rendering information, as complex data cannot be effectively organized in a flat, single level of details. In a hierarchy, different levels can abstract the content at various degrees of details and display their relationships both to sub- and parent-topics. Agglomerative approaches generate dendrograms and improve upon flat partitioning such as the k-means. However, it is generally a binary tree structure and can be unnatural for categorizing real document sets, which often do not exhibit binary divisions. It is also difficult to decide which level to cut. The divisive approaches split a chosen cluster two ways, also resulting in a binary tree structure, e.g., [19]. However, other approaches split chosen clusters n-ways, where n is fixed at each level and set *a priori*, for example in [18], $n = 9$ is used. In this case, there is an increased possibility that there may be either too few or too many clusters to represent the underlying topics. In the SOM-based hierarchical approaches, only one map can be viewed at a time e.g., GH-SOM [29] or results are reorganized and viewed as a dendrogram, e.g., TreeGCS [28].

The Dewey Decimal Classification (DDC) system, file explorers, web portals or web directories are all 1-D hierarchical, tree structures (sometimes with crossreferencing of topics). These structures typically have different numbers of subtopics at different levels and some branches may be deeper than others making it a natural, asymmetric tree. It would be advantageous to generate such a hierarchy automatically, which possesses both topology and natural tree structure. In this paper, the proposed approach, termed TOC, generates, from a set of unclassified documents, an insightful hierarchical directory of topics, while validating the dynamic structures at each level. The method can be used to categorize and organize web documents efficiently and intuitively. Topics judged similar are located closely at each level in the hierarchy, while different contents are located far apart. No further cluster identification or other post-processing is required.

B. Presentation of the Topologically Organized Content

The TOC is a set of independently spanned and hierarchically organized 1-D growing SOMs, or 1-D GCs. The preliminary work on a tree type SOM was first introduced in [43] and later enhanced by using short context [44]. The principle was to let a root chain grow until some terminating QE is reached. Then each node is tested to determine if documents clustered to it should be further clustered to a child chain. This is similar to divisive clustering where each cluster is divided into two smaller clusters at each step. However, the tree type SOM made the split as an n-way tree, where n can be different at different levels. The main limitation of this earlier method is that a stopping criterion, e.g., QE reduction ratio, has to be decided before the training. If this ratio is set too large or too small then the chain will become a flat partition or become too indiscriminative. Another weakness in this approach is that the Euclidean distance is used in sparse document space, where the cosine correlation is considered more appropriate [45].

In this paper, the TOC uses an entropy-based BIC to determine the optimum number of nodes for each GC, or subdirectory. Once the number of nodes is determined, a heuristic method is used to decide if documents clustered to a node should become a leaf (or end) node or should be further expanded into a child chain. Each child chain has a reduced vocabulary compared to its parent node, allowing it to specialize in a subarea. In this way, a tree is formed as a hierarchy of GCs. This paper also introduces a novel method to deal efficiently with sparse data vectors using the dot product, and subsequently produces a faster winner search method and weight update function. Furthermore a flexible node insertion method for the GCs is employed using both interpolation and extrapolation. Finally a new node-labeling scheme using the most descriptive terms for each node is introduced to allow a natural browsing of contents. A general diagram showing the overall system is depicted in Fig. 2.

C. Indexing and Feature Selection

In the first stage the HTML files (or other document files such as plain text, e-mail, Microsoft Word and Adobe PDF) in the dataset are crawled, text in the title tags are saved and text enclosed in HTML tags or comments are ignored. The title and remaining text is then indexed by recording successive words using a sliding window of sizes 1 to p (e.g., $p = 4$, up to four

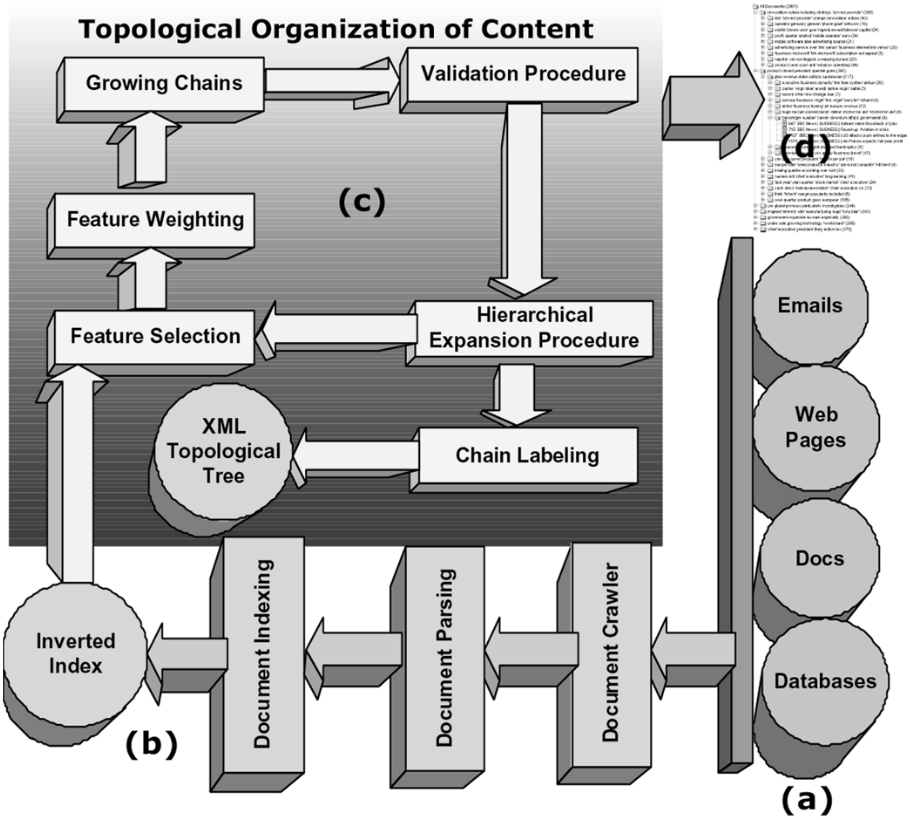


Fig. 2. General diagram of the TOC process. (a) Content from e-mails, web pages, documents or databases are collected. (b) This content is crawled, parsed into plain text and indexing to an inverted index. (c) The closed loop generates a series of chains. First the terms are selected and weighted. Then the GC algorithm is initiated and validation measures are recorded during chain growth. These validation measures are then used to determine the optimum chain size. In the hierarchical expansion, all nodes in the current chain are tested to determine if they should become leaf nodes or span further child chains. Once this is done, the nodes in the chain are labeled with representative terms and finally added in the topological tree. (d) Once all chains have been added the final XML topological tree is output.

successive words are used for each term) over the sentences, while ignoring common stop words. Once all documents are indexed, terms occurring in less than a few documents and those that occur in more than a percentage of the total documents are discarded, as these words contribute little to help discriminate between the documents. In the vector space model [45] a document is stored as $[f_1, \dots, f_n]$, where f_i is the frequency of term i and N is the total number of unique terms. Since a document vector is often sparse, each of its term frequencies is stored in a hash table as a compressed representation for efficient retrieval

$$\begin{aligned} \mathbf{x} &= [\{\gamma_1, \rho_1\}, \{\gamma_2, \rho_2\}, \dots, \{\gamma_r, \rho_r\}] \\ \gamma_k &\in \{V \mid f_{\gamma_k} \neq 0\}, \quad k = 1, 2, \dots r \end{aligned} \tag{7}$$

where $\mathbf{x}$ is a document stored as pairs: γ_k (γ_kth term in the chain's vocabulary V) and ρ_k (the weighting defined by (8)). The typical IR term frequency multiplied by the inverse document frequency [45], is used to weight the term frequency, combined with a scheme that weights more heavily the terms containing more words, and a normalization to take into account different document length

$$\rho_{ij} = \frac{\frac{\xi \cdot f_{ij}}{s} \cdot \log \frac{m}{D_j}}{\left\| \frac{\xi \cdot f_{ij}}{s} \cdot \log \frac{m}{D_j} \right\|} \tag{8}$$

where ρ_{ij} denotes the final weighting of term j in document i. $\xi = \delta(f_j)$ is a function determining the importance of term f_j, depending on how many words are in the term. s is a constant $s = x$ (single-word) $+ \cdots + x(p-$word). f_{ij} is the frequency of term j in document i, m is the total number of documents in collection, and D_j is number of document containing term j.

D. GCs

The algorithm used to grow the 1-D SOM is termed GCs and shares growing properties with the growing grid (GG) (used in the GH-SOM) and growing SOM variants, but is more suited for 1-D. A chain begins with two nodes and is growing until a validation process terminates it. At the start the weights of the two root-nodes are initialized to small random values. This is to avoid any bias to the chains, and generally SOMs are insensitive to initialization [35], [36]. In a similar way to the SOM, there are two major steps in the GC algorithm: the search for the best matching unit and the update of the winner and its neighboring nodes. At time t, an input document vector $\mathbf{x}$ is mapped to a chain consisting of n nodes. The weight vector of node $j, \mathbf{w}_j = [w_{j1}, w_{j2}, \dots, w_{jN}]^T$, is of N dimensions (where N is the total number of unique terms in the collection). The dot product between vectors $\mathbf{x}$ and $\mathbf{w}_j$ can be computed efficiently from the compressed document representation (7) as

$$S_{\text{dot}}(\mathbf{x}(t), \mathbf{w}_j) = \sum_{k=1}^{r(t)} \rho_k \cdot w_{j\gamma_k}, \quad \forall \{\gamma_k, \rho_k\} \in \mathbf{x}(t) \tag{9}$$

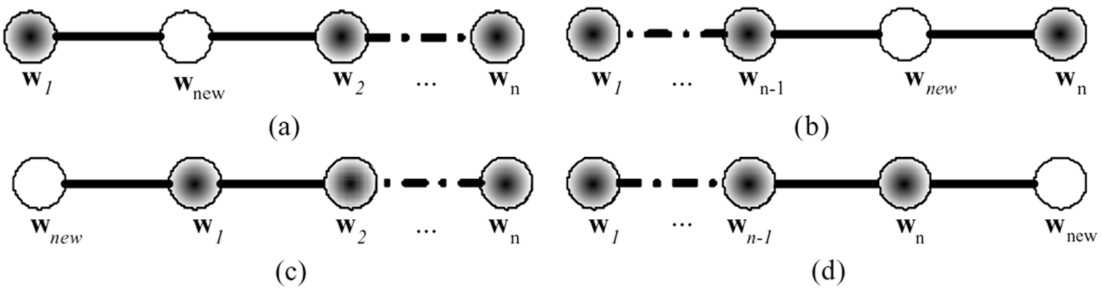


Fig. 3. Possible node insertions in the GCs.

where $r$is the number of terms in document $\mathbf{x}(t)$ and ρ_k is the value given by (8) of the term in the current document $\mathbf{x}(t)$. The method iterates through the 1 to r dimensions of the document vector rather than 1 to L, the number of terms in the chain's vocabulary where $r \ll L$ in almost all cases. The best matching unit $c(\mathbf{x})$ is the node with the maximum dot product amongst nodes j and document vector $\mathbf{x}(t)$

$$c(\mathbf{x}) = \arg\max_j \{S_{\text{dot}}(\mathbf{x}(t), \mathbf{w}_j)\}, \quad j = 1, 2, \ldots n \qquad (10)$$

where n is the current number of nodes. Once the winner node $c(\mathbf{x})$ is found the neighboring weights are updated using

$$\mathbf{w}_j(t+1) = \frac{\mathbf{w}_j(t) + \alpha(t) h_{j,c(x)}(t)\mathbf{x}(t)}{\|\mathbf{w}_j(t) + \alpha(t) h_{j,c(x)}(t)\mathbf{x}(t)\|} \qquad (11)$$

where $\alpha(t)$ is the monotonically decreasing learning rate and $h_{j,c(x)}(t)$ the neighborhood function, typically a Gaussian kernel. In order to reduce the noise from the sparse zero terms in the document vectors, the weight smaller than a threshold (e.g., 0.000 01) are set to zero. During training documents are presented in a random order.

E. Node Insertion in the GCs

Each node has a counter for its (winning) activities, which is reinitiated after each new node insertion. A measure termed average similarity (AvgSim) is also recorded during training

$$\text{AvgSim} = \frac{1}{n}\sum_{j=1}^{n} \frac{1}{m_j} \sum_{i=1}^{m_j} (S_{\text{dot}}(\mathbf{x}_i, \mathbf{w}_j)) \qquad (12)$$

where n is the current number of nodes in the chain for $m_j \neq 0$ where $2 \leq n \leq n_{\max}$ and $n_{\max}$ is the maximum number of nodes used in the validation. m_j is the number of documents clustered in a particular node $j, \mathbf{x}_i$ are the documents i such that $c(\mathbf{x}_i) = j$ and $\mathbf{w}_j$ is the weight of node j. If no document is clustered to a particular node j the dot product has a value of 0. If the chain is judged to have sufficiently converged shown by stabilization of AvgSim, then a new node is inserted next to the node with the highest number of wins. The new node is added in the proximity of the node with the largest counter and to its right or left, depending on which situation will lead to the highest AvgSim increase of the chain. The weight of the new node is initialized using interpolation between these two nodes like in the two cases shown in Fig. 3(a), (b) $\mathbf{w}_{\text{new}} = (\mathbf{w}_i + \mathbf{w}_{i+1})/2$. If the node with the largest counter is found to be on a boundary then a new node is either interpolated or extrapolated, depending on which will increase the overall chain AvgSim. The nodes inserted at boundaries have their weights initialized through extrapolation by (13) as shown in Fig. 3(c), (d). If the difference is negative then the weights are set to zero, as no terms in the document vectors are negative

$$\mathbf{w}_{\text{new}} = \begin{cases} 0 & \text{if } (2\mathbf{w}_1 - \mathbf{w}_2) \leq 0 \\ (2\mathbf{w}_1 - \mathbf{w}_2) & \text{otherwise.} \end{cases}$$
$$\mathbf{w}_{\text{new}} = \begin{cases} 0 & \text{if } (2\mathbf{w}_n - \mathbf{w}_{n-1}) \leq 0 \\ (2\mathbf{w}_n - \mathbf{w}_{n-1}) & \text{otherwise.} \end{cases} \qquad (13)$$

During the growth and just before a new node is inserted, the weights of all the existing nodes in the chain are stored along with the validation measure for the current number of nodes n. The growth process is terminated when the validation has chosen the correct number of nodes—the details are described next.

F. GCs Validation

The validation is essential for determining the optimum number of nodes at each level, thus, an optimal content hierarchy. Rather than performing costly full growth (to one document one cluster) first and then pruning it like in machine learning decision trees, or simultaneous addition and removal of nodes until a threshold is reached (like in ART or GCS), the chain is allowed to grow to a specified number of nodes $n_{\max}$. This is analogous to the k-means mean-square-validation where the optimum k is searched for where $k = 1, \ldots, n_{\max}$.

The idea of using validation is to discover or choose the natural number of nodes or topics. Generally one seeks to maximize the intracluster (within the cluster) and minimize the intercluster (between clusters) similarities. In real world applications, this is usually not feasible since many outliers, noisy points or overlapping clusters are present in the data. This measure is already given by AvgSim and optimized by the GCs. The proposed validation is based on the documents entropy instead of simple centroid. The validation measure penalizes larger number of clusters, which encourages a flat partition. This measure is applied locally to a particular chain, taking into account its reduced vocabulary. The first step in the validation is to calculate the distribution of terms in each cluster using the probability that a term t_i belongs to cluster C_j as

$$p(t_i \mid C_j) = \frac{c(t_i \mid C_j)}{\sum_{t'_i \in C_j} c(t'_i \mid C_j)} \qquad (14)$$

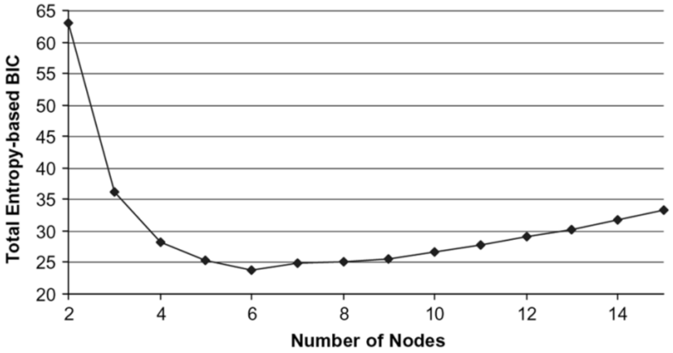


Fig. 4. Average of five runs of the proposed entropy-based BIC validation to the number of nodes, where $\tau = 6$.

where $c(t_i \,|\, C_j)$ is the count of occurrence of t_i in cluster C_j. Then the entropy H of cluster C_j is defined as

$$H(C_j) = -\sum_i p(t_i \,|\, C_j) \log p(t_i \,|\, C_j). \tag{15}$$

The total entropy is the normalized and weighted sum of entropies

$$H_t(n) = \frac{1}{m} \sum_{j=1}^{n} m_j \cdot H(C_j) \tag{16}$$

where m is the total number of documents, n the current number of clusters and m_j is the size of the cluster. Choosing n for which H_t is lowest favors large numbers of clusters since entropy decreases with an increase in the number of clusters. Hence, Schwarz's BIC (6) is used to penalize the increasing complexity, as more nodes are added to a chain. The proposed validation criterion estimates the most suitable number of nodes τ by

$$\tau = \arg\min_n \left\{ H_t(n) + \frac{1}{2} n \log m \right\}, \quad n = 2, \ldots, n_{\max} \tag{17}$$

where m is the number of documents. An example of the entropy-based BIC validation curve is given in Fig. 4. The average of five runs (all indicated $\tau = 6$) was taken where weights were initialized and documents selection was performed randomly. It is clear that the proposed validation method indicates six clusters, the resulting chain is shown in Fig. 1. The chain with τ nodes is restored, added to the hierarchy and its nodes are used for further clustering (of nonleaf nodes) or final presentation (leaf nodes).

G. Self-Organizing Growing Trees Process

Once the root-chain has been trained and the optimum number of nodes τidentified, each node is tested to see whether documents clustered to it need further expansion into a child chain Fig. 2. One heuristic approach is to allow a minimum number of documents in each topic node. Once the local dictionary of terms is created, another test checks the number of term in chain vocabulary. If the vocabulary only contains few terms this means that the documents may already be similar, the parent-node is sufficiently specialized and there are insufficient terms for further expansion. The final test uses reduced vocabulary with a cluster tendency heuristic. This aims to test if a set of documents contains random documents with no or little relations or if there are strong underlying clusters. These tests are useful for deciding whether one should perform the further clustering. Since most clustering algorithm will find clusters even in random data and many validation measures cannot validate single clusters. The cluster density tendency method is given in [46]. If any of these methods fail then the parent-node of the candidate child chain becomes a leaf node.

If the parent-node spans a child, then all documents mapped to the parent-node form a new subset of documents with reduced vocabulary. After the child chain is created the feature selection is performed again as explained in Section III-C and applied along with (8). The value of ρ is calculated from the original term frequency and not the parent-node's document vectors (as these are weighted). This process is important because the parent-node has already used specific feature terms to cluster the documents and some of these terms in the child documents may no longer be discriminative (e.g., terms originally occurring in 5% of the document in the parent-set may occur in 100% of documents the child set). It is shown in Section IV, that about 50% of the terms can be discarded from one level to the next. If a child chain is added then it will start with two nodes. In this way, each new addition to the tree contributes to the specialization of the vocabulary and topics in child levels.

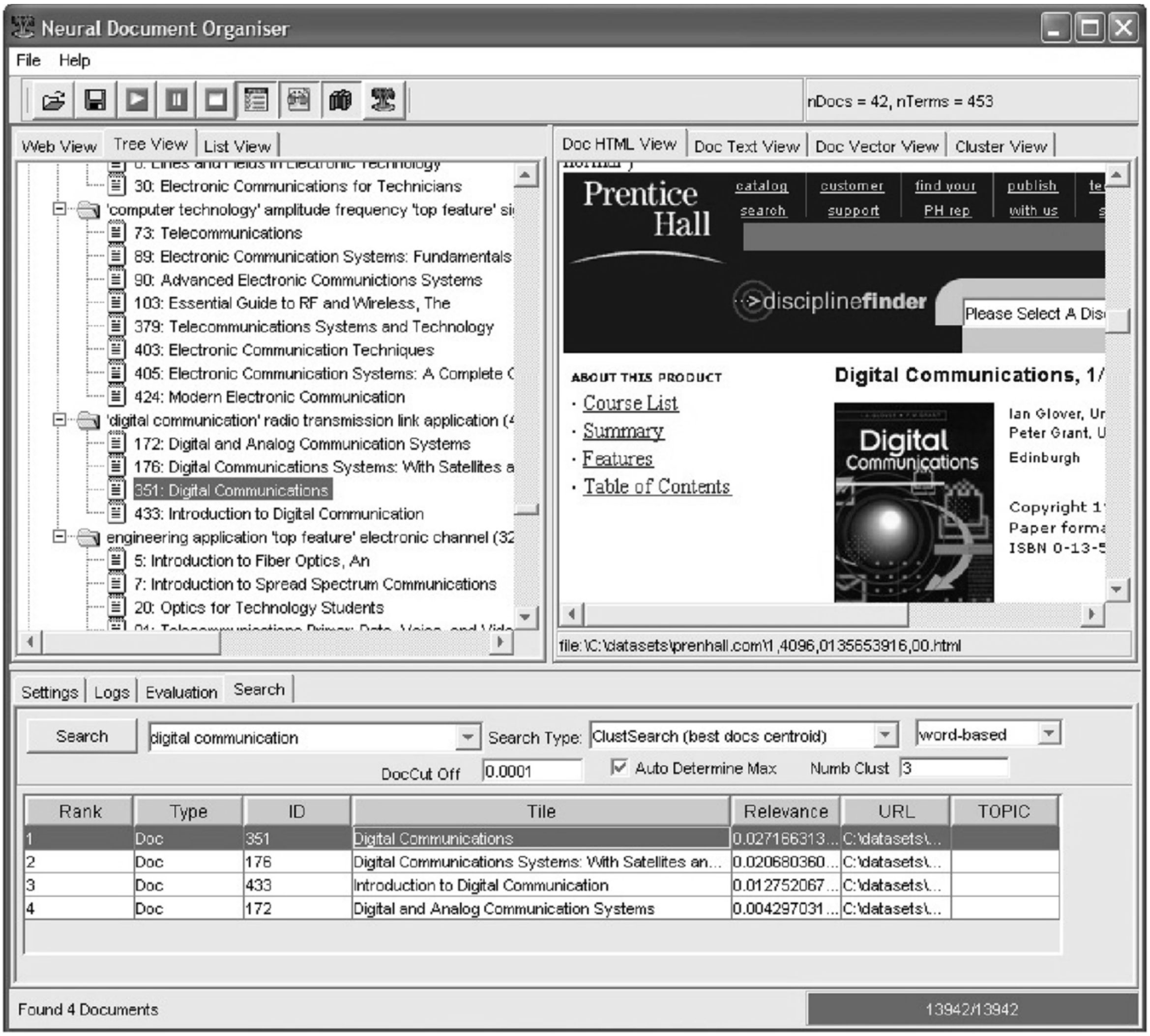
H. Visualization and Implementation

An ordered tree is a natural and intuitive way to present document contents and their structures, and it is more helpful and advantageous for understanding and navigating documents than other presentations such as dendrograms (generated from typical hierarchical clustering), flat partitioning and 2-D grids.

The TOC method does not require any further processing or manual intervention, to obtain topic hierarchy. Each cluster is labeled with a number of representative key words. First, all the terms judged statistically irrelevant are removed from the possible terms for labeling. For the remaining terms, the term score ts_{ji} for terms i in node j is calculated for document k

$$\mathrm{ts}_{ji} = \sum_{k=1}^{m_j} \rho_k(i) w_{j\gamma(i)}, \quad i = 1, 2, \ldots r_k. \tag{18}$$

It was found that simultaneously ranking the terms by frequency of occurrence and $\mathbf{ts}_i$ (as primary key) provided high-quality and understandable cluster labels. The first few terms [e.g., (5)] with the highest values are selected to label the cluster. The label summarizes the content of the cluster and allows the user to identify topics quickly at a particular level or related areas. An advantage of this labeling scheme is that the same terms used by the GC to cluster the documents are also used to label the nodes. This reflects the importance of key terms in the weights of the nodes. Quotation marks are used to distinguish the terms with multiple words from those with single words. A bottom-up heuristic method is used to avoid duplication of terms (e.g., term "economic theory" and words "economic" "theory") and allows more precise labeling. Concise and brief are essential as only limited text space is available for the label.



30: Electronic Communications for Technicians
'computer technology' amplitude frequency 'top feature' si
73: Telecommunications
89: Electronic Communication Systems: Fundamentals
90: Advanced Electronic Communictions Systems
103: Essential Guide to RF and Wireless, The
379: Telecommunications Systems and Technology
403: Electronic Communication Techniques
405: Electronic Communication Systems: A Complete C
424: Modern Electronic Communication
'digital communication' radio transmission link application (4
172: Digital and Analog Communication Systems
176: Digital Communications Systems: With Satellites a
351: Digital Communications
433: Introduction to Digital Communication
engineering application 'top feature' electronic channel (32
5: Introduction to Fiber Optics, An
7: Introduction to Spread Spectrum Communications
20: Optics for Technology Students

file:\C:\datasets\prenhall.com\1,4096,0135653916,00.html

Search: digital communication
Search Type: ClustSearch (best docs centroid)
word-based
DocCut Off 0.0001
☑ Auto Determine Max
Numb Clust 3

Rank	Type	ID	Tile	Relevance	URL	TOPIC
1	Doc	351	Digital Communications	0.027166313...	C:\datasets\...	
2	Doc	176	Digital Communications Systems: With Satellites an...	0.020680360...	C:\datasets\...	
3	Doc	433	Introduction to Digital Communication	0.012752067...	C:\datasets\...	
4	Doc	172	Digital and Analog Communication Systems	0.004297031...	C:\datasets\...	

Found 4 Documents
13942/13942

Fig. 5. Screen shot of the developed Java application prototype.

A full application, termed Neural Document Organizer application, with integrated crawler and browser has been implemented in Java for efficiency and crossplatform support. The settings for preprocessing and clustering can be customized using the graphical user interface (GUI) as shown in Fig. 5. The program takes a set of HTML/text documents and generates a topological tree view of topics. The generated tree view is a JTree from the Java Swing GUI class. The Java JTree is mouse sensitive, e.g., clicking on a branch expands or collapses it. Selecting a document displays its content in the right content window, together with terms/weightings at the bottom. Clicking on a chain-node reveals information about the cluster (terms, weight, AvgSim, validation measures, etc.). The application can also generate a client-side JavaScript tree view useful for web administrators to create a dynamic table of contents. In addition, the application generates XML data allowing interoperability with other applications, web services or databases. This is an important feature in content management as the structure can be used by both humans and automated services [1].

IV. Results and Discussions

This section presents the results obtained from the proposed TOC, bisecting k-means, SOM and GH-SOM methods on various datasets for comparison. The datasets used are summarized in Table I. The web pages contain descriptions of books or recent news articles. Web pages were chosen in this application as they are one of the most wide spread forms of content on the Internet.

TABLE I
Datasets Using in the Experiments

Dataset	No. of documents	No. of terms	Source
A	618	7163	http://www.prenhall.com
B	1097	12171	http://www.aw.com
C	2001	17398	http://news.bbc.co.uk

A. Datasets

Dataset A consists of 618 web pages describing books published by Prentice Hall. These textual documents are of variable content and length, and contain topics from *accounting*, *computing*, *health care*, *business* to *engineering*. Dataset B contains book descriptions from Addison Wesley and Benjamin Cummings. The document content is also variable. At minimum

TABLE II
PERFORMANCE COMPARISON BETWEEN SOM, GH-SOM, AND TOC FOR DATASET A

Level	TOC	GH-SOM	SOM
Root Level	258.578s	2805.109s	16.518h
Average Level 1	73.353s	435.295s	n/a
Average Level 2	20.382s	65.828s	n/a
Total Time	0.220h	1.822h	16.518h

it may be a title and possibly including table of contents, descriptions or features. The dataset is related to *medical* and *life sciences*, *mathematics*, *computer science*, *physics*, and *economics*. Dataset C is a set of documents collected from the BBC news website. They are full news articles covering *business*, *economics*, *finance*, *stock markets*, and *e-commerce*, each of which contains descriptive HTML title tags. The news articles lead to a very large number of indexing terms. This dataset is complex as the topic area is much narrower, focusing on business related issues, rather than general and clearly separated such as in datasets A and B.

All datasets are different from one another, especially datasets A and B, which contain more technical terms and may not contain full sentences, whereas dataset C uses common and journalistic English. These unlabeled documents represent a challenge for the clustering algorithms as they deal with a variety of overlapping topics. The document structures by nature are hierarchical so flat partitioning algorithms would not be able to represent them efficiently. Another difficulty is that the language used for each topic is very specialized, leading to the need of efficient methods for dealing with a sparse vocabulary (see Table I).

B. Quantitative Comparisons

All four methods have been applied to the same datasets with the same preprocessing techniques. SOM, GH-SOM, and TOC methods used the same dot product and speedups as described in Section III-D. The indexing and initial feature reduction methods were the same in all three approaches. The total times taken using a 1.2 GHz AMD processor for both clustering and labeling of the three SOM-based approaches for datasets A to C, are listed in Table II. The SOM took longer than the TOC, due to the larger number of nodes, winner searches and neighborhood updates. The TOC was also much faster than the GH-SOM as shown in Table II. In addition to improved computational performance, the topological tree is easy to navigate at various levels simultaneously. This is not possible with 2-D representations such as the SOM, GCS or GH-SOM.

One of the advantages of a hierarchy is its ability to represent concepts from general in the root-level to specific in the leaf levels. The reduction and specialization in vocabulary between levels is shown in Fig. 6, using the TOC algorithm with dataset A, B, and C. On average the number of terms is halved from one level to the next, leading to faster training and vocabulary specialization in child chains.

The four resulting structures have been compared in terms of depth and average number of nodes, as these are important features for browsing. Table III shows the depth of each structure for each of the datasets. The deepest structure was the binary tree generated from the bisecting k-means, however this depth was user defined. The lowest was the SOM, which does not form a hierarchy and results in flat partitions. The depth of the TOC and GH-SOM were determined dynamically and were roughly equivalent. Two to three levels in depth can intuitively be browsed.

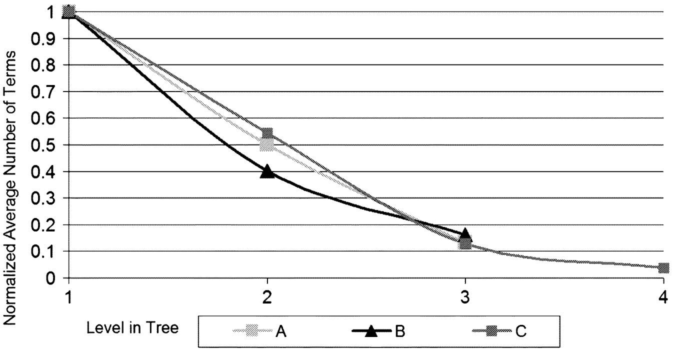


Fig. 6. Normalized average number of terms at each level for datasets A, B, and C.

TABLE III
NUMBER OF LEVELS IN GENERATED STRUCTURES

Dataset	TOC	SOM	GH-SOM	Bisecting k-means
A	3	1	3	8
B	3	1	3	8
C	4	1	3	8

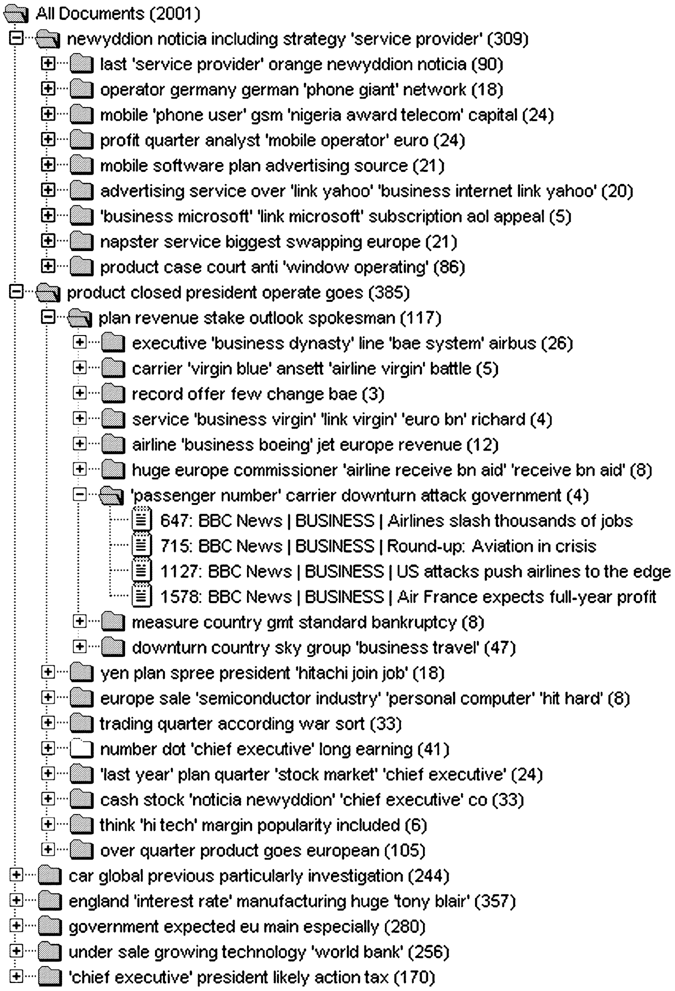
TABLE IV
AVERAGE NUMBER OF NODES PER MAP/CHAIN FOR ALL LEVELS

Dataset	TOC	SOM	GH-SOM	Bisecting k-means
A	6.33	36	9.93	2
B	6.23	49	13.31	2
C	7.42	49	12.30	2

Another important measure is the number of nodes at each level. Table IV compares different resulting structures for all three datasets. For the SOM there were too many nodes, hence, requiring many labels and further higher level labels may be required, for example, the SOMs had to be manually segmented for datasets A, B, and C. For the bisecting k-means, there were too fewer nodes per level requiring the structure to be very deep (see Table III). The TOC and GH-SOM both had a more adequate number of nodes compared to the SOM and bisecting k-means.

C. Visual Comparisons

The results of the TOC, SOM, GH-SOM, and bisecting k-means are shown and empirically compared on advantages and drawbacks at visualizing documents. For the 2-D maps the document titles had to be omitted in the topic visualization, as they would not fit in the space provided. Details of the clusters are normally displayed on separate pages. Due to the space considerations only the clustering results, instead of the entire



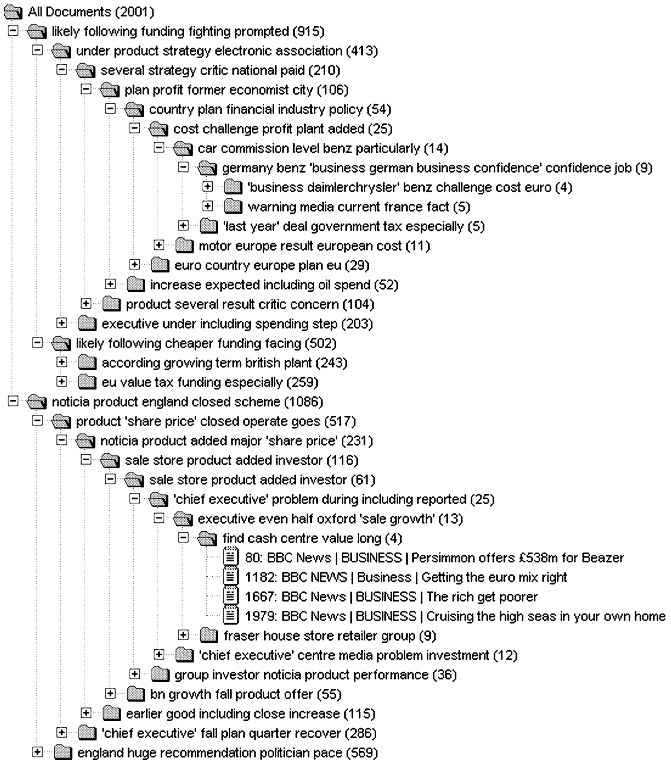
```
All Documents (2001)
  newyddion noticia including strategy 'service provider' (309)
    last 'service provider' orange newyddion noticia (90)
    operator germany german 'phone giant' network (18)
    mobile 'phone user' gsm 'nigeria award telecom' capital (24)
    profit quarter analyst 'mobile operator' euro (24)
    mobile software plan advertising source (21)
    advertising service over 'link yahoo' 'business internet link yahoo' (20)
    'business microsoft' 'link microsoft' subscription aol appeal (5)
    napster service biggest swapping europe (21)
    product case court anti 'window operating' (86)
  product closed president operate goes (385)
    plan revenue stake outlook spokesman (117)
      executive 'business dynasty' line 'bae system' airbus (26)
      carrier 'virgin blue' ansett 'airline virgin' battle (5)
      record offer few change bae (3)
      service 'business virgin' 'link virgin' 'euro bn' richard (4)
      airline 'business boeing' jet europe revenue (12)
      huge europe commissioner 'airline receive bn aid' 'receive bn aid' (8)
      'passenger number' carrier downturn attack government (4)
        647: BBC News | BUSINESS | Airlines slash thousands of jobs
        715: BBC News | BUSINESS | Round-up: Aviation in crisis
        1127: BBC News | BUSINESS | US attacks push airlines to the edge
        1578: BBC News | BUSINESS | Air France expects full-year profit
      measure country gmt standard bankruptcy (8)
      downturn country sky group 'business travel' (47)
    yen plan spree president 'hitachi join job' (18)
    europe sale 'semiconductor industry' 'personal computer' 'hit hard' (8)
    trading quarter according war sort (33)
    number dot 'chief executive' long earning (41)
    'last year' plan quarter 'stock market' 'chief executive' (24)
    cash stock 'noticia newyddion' 'chief executive' co (33)
    think 'hi tech' margin popularity included (6)
    over quarter product goes european (105)
  car global previous particularly investigation (244)
  england 'interest rate' manufacturing huge 'tony blair' (357)
  government expected eu main especially (280)
  under sale growing technology 'world bank' (256)
  'chief executive' president likely action tax (170)
```

Fig. 7. Taxonomy of dataset C using the proposed TOC method.

screen shot of the developed Neural Document Organizer application (e.g., Fig. 5), are displayed in the subsequent figures and only generated structures for dataset C are given.

The purpose for comparing the TOC to 2-D SOM-based methods was to demonstrate that the topology in tree structures can improve navigation and visualization. The bisecting k-means was also compared as it is a representative example of hierarchical document clustering methods and has been shown to outperform agglomerative hierarchical document clustering [17]. The bisecting k-means is a divisive method, which generates a dendrogram representation. To improve the labeling and cluster hierarchy generation speed, the bisecting k-means algorithm has been modified so that the feature terms passing from one parent cluster to two child clusters are reduced. This is similar to the TOC, where the documents clustered to a node are expanded into a child chain with reduced vocabulary.

Fig. 7 shows a topological tree generated using the TOC. From top to bottom, resulting topics are *telecom* and *communications*, *airlines* and *high tech*, *automobiles*, *manufacturing*, *government*, *emerging markets*, and *senior executive* and *management*. The topology reflects strong links between *government*, and *interest rates* and *manufacturing*. It also merges strongly related topics and expands them into a child chain, e.g., *service providers* such as *mobile phones*, *Yahoo* and *Napster* (peer-2-peer). These labels help illustrate the content of the documents without browsing the document title or details.



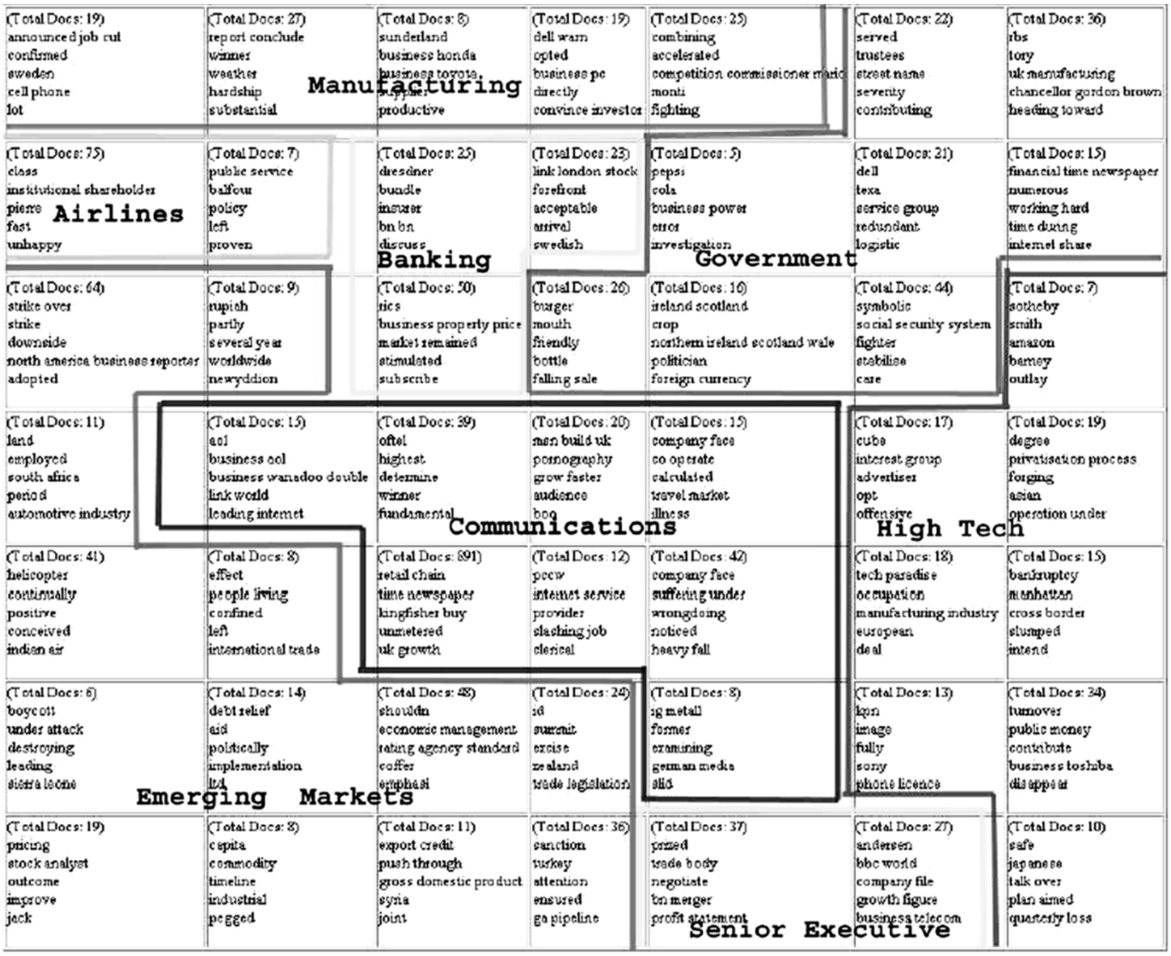
```
All Documents (2001)
  likely following funding fighting prompted (915)
    under product strategy electronic association (413)
      several strategy critic national paid (210)
        plan profit former economist city (106)
          country plan financial industry policy (54)
            cost challenge profit plant added (25)
              car commission level benz particularly (14)
                germany benz 'business german business confidence' confidence job (9)
                  'business daimlerchrysler' benz challenge cost euro (4)
                  warning media current france fact (5)
                'last year' deal government tax especially (5)
              motor europe result european cost (11)
            euro country europe plan eu (29)
          increase expected including oil spend (52)
        product several result critic concern (104)
      executive under including spending step (203)
    likely following cheaper funding facing (502)
      according growing term british plant (243)
      eu value tax funding especially (259)
  noticia product england closed scheme (1086)
    product 'share price' closed operate goes (517)
      noticia product added major 'share price' (231)
        sale store product added investor (116)
          sale store product added investor (61)
            'chief executive' problem during including reported (25)
              executive even half oxford 'sale growth' (13)
                find cash centre value long (4)
                  80: BBC News | BUSINESS | Persimmon offers £538m for Beazer
                  1182: BBC NEWS | Business | Getting the euro mix right
                  1667: BBC News | BUSINESS | The rich get poorer
                  1979: BBC News | BUSINESS | Cruising the high seas in your own home
                fraser house store retailer group (9)
              'chief executive' centre media problem investment (12)
            group investor noticia product performance (36)
          bn growth fall product offer (55)
        earlier good including close increase (115)
      'chief executive' fall plan quarter recover (286)
    england huge recommendation politician pace (569)
```

Fig. 8. Taxonomy of dataset C using the bisecting k-means method.

The binary tree generated using the bisecting k-means method is shown in Fig. 8. The upper section deals with *automobiles* and *high tech*, and lower section *senior executives* and *government*. The topics are much more difficult to infer than in Fig. 7. The main differences are that there are more labels, increased depth and lack of topology in the binary tree compared to the topological one.

In Fig. 9, is the resulting 2-D map generated using the SOM on dataset C. The major topics are *telecom*, *airlines*, *high tech*, *manufacturing*, *government*, *emerging markets*, and *senior executive*. Some topics are merged together like *automobile* and *manufacturing*. The map is difficult to manually segment than other datasets, as the topics are less clearly separated or heavily overlapping. Once manually segmented, the comprehension and clarity of the map is improved. However, some of the spatial relations between nodes are still difficult to explain and apprehend as nodes have up to eight neighbors. The map is also unable to abstract different levels of details, e.g., the lower node in communications contains 891 documents.

Fig. 10 shows the results of applying the GH-SOM on dataset C. On the top level of the branch, the map shows in the upper left the themes relating to *airlines* and *manufacturing*, to the upper right *government*, the lower left *emerging markets* and lower right *communication* and *telecom*. The middle map expands on the topics of *airways* and *emerging markets*. The child map expands on the topic of *airlines*. The hierarchical relations between parent nodes and child maps allow an abstraction of topics. However, when a topic spans more than one node then



(Total Docs: 19) announced job cut confirmed sweden cell phone lot	(Total Docs: 27) report conclude winner weather hardship substantial	(Total Docs: 8) sunderland business honda business toyota supplier productive	(Total Docs: 19) dell warn opted business pc directly convince investor	(Total Docs: 25) combining accelerated competition commissioner mario monti fighting	(Total Docs: 22) served trustees street name seventy contributing	(Total Docs: 36) rbs tory uk manufacturing chancellor gordon brown heading toward
(Total Docs: 75) class institutional shareholder pierre fast unhappy	(Total Docs: 7) public service balfour policy left proven	(Total Docs: 25) dresdner bundle insurer bn bn discuss	(Total Docs: 23) link london stock forefront acceptable arrival swedish	(Total Docs: 5) pepsi cola business power error investigation	(Total Docs: 21) dell texa service group redundant logistic	(Total Docs: 15) financial time newspaper numerous working hard time during internet share
(Total Docs: 64) strike over strike downside north america business reporter adopted	(Total Docs: 9) rupiah partly several year worldwide newyddion	(Total Docs: 50) tics business property price market remained stimulated subscribe	(Total Docs: 26) burger mouth friendly bottle falling sale	(Total Docs: 16) ireland scotland crop northern ireland scotland wale politician foreign currency	(Total Docs: 44) symbolic social security system fighter stabilise care	(Total Docs: 7) sotheby smith amazon barney outlay
(Total Docs: 11) land employed south africa period automotive industry	(Total Docs: 15) aol business aol business wanadoo double link world leading internet	(Total Docs: 39) oftel highest determine winner fundamental	(Total Docs: 20) msn build uk pornography grow faster audience boo	(Total Docs: 15) company face co operate calculated travel market illness	(Total Docs: 17) cube interest group advertiser opt offensive	(Total Docs: 19) degree privatisation process forging asian operation under
(Total Docs: 41) helicopter continually positive conceived indian air	(Total Docs: 8) effect people living confined left international trade	(Total Docs: 891) retail chain time newspaper kingfisher buy unmetered uk growth	(Total Docs: 12) pccw internet service provider slashing job clerical	(Total Docs: 42) company face suffering under wrongdoing noticed heavy fall	(Total Docs: 18) tech paradise occupation manufacturing industry european deal	(Total Docs: 15) bankruptcy manhattan cross border slumped intend
(Total Docs: 6) boycott under attack destroying leading sierra leone	(Total Docs: 14) debt relief aid politically implementation ltd	(Total Docs: 48) shouldn economic management rating agency standard coffer emphasi	(Total Docs: 24) id summit excise zealand trade legislation	(Total Docs: 8) ig metall former examining german media slid	(Total Docs: 13) kpn image fully sony phone licence	(Total Docs: 34) turnover public money contribute business toshiba disappear
(Total Docs: 19) pricing stock analyst outcome improve jack	(Total Docs: 8) capita commodity timeline industrial pegged	(Total Docs: 11) export credit push through gross domestic product syria joint	(Total Docs: 36) sanction turkey attention ensured ga pipeline	(Total Docs: 37) prized trade body negotiate tm merger profit statement	(Total Docs: 27) andersen bbc world company file growth figure business telecom	(Total Docs: 10) safe japanese talk over plan aimed quarterly loss

Fig. 9. 2-D 7 × 7 SOM for dataset C with manual segmentation.

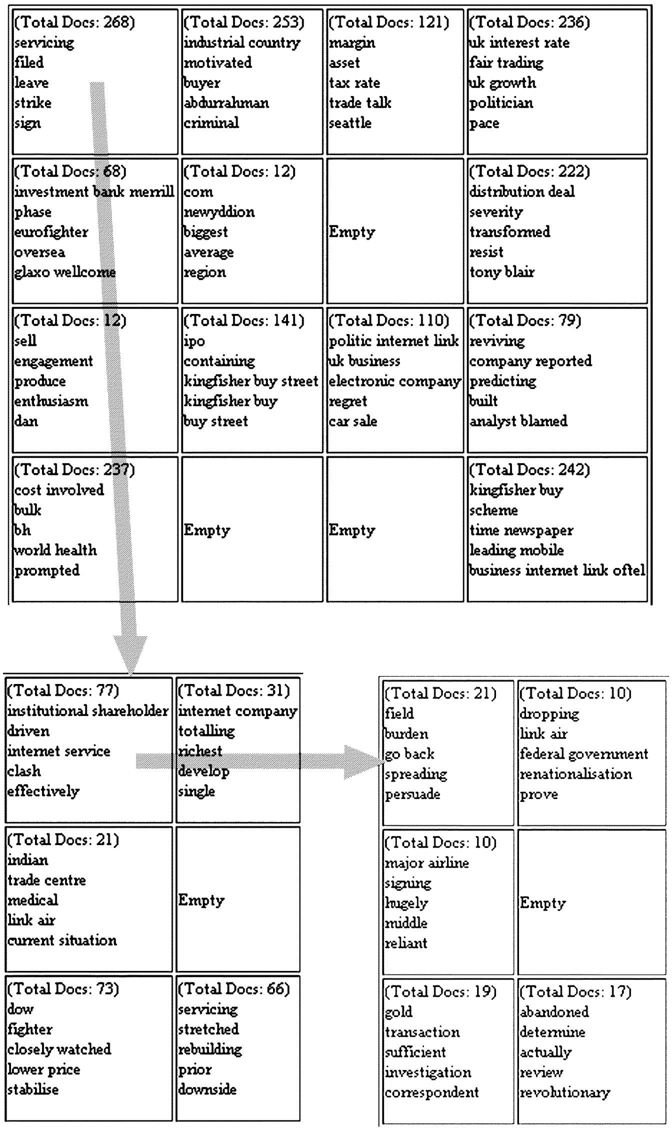
it is subdivided into different child maps. This limits the intuitiveness, as it is only possible to view one map at a time, and it is not possible to visualize different levels simultaneously in the hierarchy of maps. This unavoidably restricts the practicality of the 2-D SOM-based methods especially when dealing with huge datasets. The topological tree representation, however, does allow such viewing and representation, and will not be hampered by the larger scale of the dataset.

V. Conclusion and Future Work

The proposed TOC method produces a useful hierarchy of topics that is automatically labeled and validated at each level. The main advantages of the approach are the validation measure, scalability, and topology representation. The TOC provides abstractions of content at various levels. At each level the organization is optimally validated. The advantage of local rather than global optimum is that entire reclustering is not required as in many other methods, for examples, in [21] entire clustering is repeated, and in [47] the best of many runs is taken. This also makes the TOC fully scalable and adaptive for large and/or distributed applications. The validation measure takes into account local specialized topics in each chain. In addition, no tree pruning is required. At each level the optimum number of nodes has been chosen, saving on memory storage and post-processing. The estimation of the best number of nodes in the GC, through validation adds very little overhead. The validation measure is only calculated after each node insertion.

The TOC uses similar splitting or divisive clustering principles, amongst which is the scalability compared to the agglomerative method [19], as well as the advantages of the SOM, i.e., topology preservation [26], convergence [35] and scalability [10]. The TOCs chains use much fewer nodes compared to a 2-D SOM. The GC only inserts a single node, rather than a grid, at a time, giving advantage that validation can be monitored simultaneously. Also a fast winner search, is used by computing only the terms in the document vector (instead of the weight) for all nodes.

The TOC algorithm generates a tree, which is easy to navigate at various levels at same time—this is not possible with 2-D representations such as SOM and GH-SOM. The tree can be used directly for content management and exploration, unlike other representations primarily aimed at retrieval. Dendrograms used for browsing large datasets need to be cut either horizontally or at branch level. This is an arbitrary process and usually involves prefixed thresholds. The tree structural visualization can potentially display, store and access a great number of topics with ease, with the ability to show and hide any branch in the tree. The topology of chains helps the user find related topics at each level naturally and the meaningful labels help interpret their similarities.



(Total Docs: 268) servicing filed leave strike sign	(Total Docs: 253) industrial country motivated buyer abdurrahman criminal	(Total Docs: 121) margin asset tax rate trade talk seattle	(Total Docs: 236) uk interest rate fair trading uk growth politician pace
(Total Docs: 68) investment bank merrill phase eurofighter oversea glaxo wellcome	(Total Docs: 12) com newyddion biggest average region	Empty	(Total Docs: 222) distribution deal severity transformed resist tony blair
(Total Docs: 12) sell engagement produce enthusiasm dan	(Total Docs: 141) ipo containing kingfisher buy street kingfisher buy buy street	(Total Docs: 110) politic internet link uk business electronic company regret car sale	(Total Docs: 79) reviving company reported predicting built analyst blamed
(Total Docs: 237) cost involved bulk bh world health prompted	Empty	Empty	(Total Docs: 242) kingfisher buy scheme time newspaper leading mobile business internet link oftel

(Total Docs: 77) institutional shareholder driven internet service clash effectively	(Total Docs: 31) internet company totalling richest develop single
(Total Docs: 21) indian trade centre medical link air current situation	Empty
(Total Docs: 73) dow fighter closely watched lower price stabilise	(Total Docs: 66) servicing stretched rebuilding prior downside

(Total Docs: 21) field burden go back spreading persuade	(Total Docs: 10) dropping link air federal government renationalisation prove
(Total Docs: 10) major airline signing hugely middle reliant	Empty
(Total Docs: 19) gold transaction sufficient investigation correspondent	(Total Docs: 17) abandoned determine actually review revolutionary

Fig. 10. Branch from the hierarchy generated on dataset C using the GH-SOM.

To scale the system further, feature reduction methods such as random projection [10] can be used to further reduce the number of terms. The calculation of normalizing denominator in the weight update rule, (11), can also been improved to speed up the process. On some occasions a couple of documents shifted from one node to a neighboring node. This is due to the random presentation of documents and initialization weights. This effect can be reduced through the use of cyclic ordering of documents e.g., [28]. However, the principle eigenvector in the document space can be used to initialize the chains and batch training can also be used as in the WEBSOM [10].

The automated labeling produces consistent labels compared with human indexing, which heavily depends on indexers' knowledge and experience. The labels provide the user with an overview of unknown topics for facilitating exploration and browsing. The root-level labels may be general, because only shared vocabulary amongst all document clusters is used for labeling. A better way to label the highest-level sets would be to infer terms using external knowledge. On many occasions the topology can be helpful in overcoming insensitive labels as neighboring topics are closely related and neighboring labels are complimentary. Future work will include using more advanced feature selection methods such as the latent semantic analysis [48] or WordNet[3] to enhance the associations between terms. Future work will also adapt the TOC method and generated topological trees for ontology creation for the semantic web or databases and fully integrate with web portals.

Acknowledgment

The authors are grateful to the reviewers for their helpful comments.

References

[1] S. Goodwin and R. Vidgen, "Content, content, everywhere... time to stop and think? The process of web content management," *Comput. Control Eng. J.*, vol. 13, no. 2, pp. 66–70, 2002.

[2] D. O'Meara, "Buried in documents?," *Eng. Manage. J.*, vol. 10, no. 5, pp. 241–243, 2000.

[3] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. Archibald, and X. Liu, "Learning approaches for detecting and tracking news events," *IEEE Intell. Syst.*, vol. 14, no. 4, pp. 32–43, 1999.

[4] L. Manevitz and M. Yousef, "Document classification on neural networks using only positive examples," in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, vol. 34, 2000, pp. 304–306.

[5] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization*, 1998.

[6] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. Machine Learning: ECML-98 10th Eur. Conf. Machine Learning*, Chemnitz, Germany, 1998, pp. 137–142.

[7] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[8] D. Cutting, D. Karger, J. Pederson, and J. Tukey, "Scatter/gather: A cluster-based approach to browsing large document collections," in *Proc. 15th Int. ACM/SIGIR Conf. Research and Development in Information Retrieval*, Copenhagen, Denmark, 1992, pp. 318–329.

[9] C. J. Van Rijsbergen, *Information Retrieval*, 2nd ed. London, U.K.: Butterworth, 1979.

[10] K. Lagus, S. Kaski, and T. Kohonen, "Mining massive document collections by the websom method," *Inf. Sci.*, vol. 163, no. 1–3, pp. 135–156, 2004.

[11] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration," in *Proc. 21st Annu. Int. SIGIR Conf. Research and Development in Information Retrieval*, 1998, pp. 46–54.

[12] R. Freeman, "Web document search, organization, and exploration using self-organizing neural networks," Ph.D. dissertation, School of Elect. and Electron. Eng., Univ. Manchester, Manchester, U.K., 2004.

[13] F. Meziane and Y. Rezgui, "A document management methodology based on similarity contents," *Inf. Sci.*, vol. 158, pp. 15–36, 2003.

[14] E. Rasmussen and P. Willett, "Non-hierarchic document clustering using the icl distributed array processor," in *Proc. 10th Annu. Int. ACMSIGIR Conf. Research and Development in Information Retrieval*, New Orleans, LA, 1987, pp. 132–139.

[15] M. Sahami, S. Yusufali, and M. Q. W. Baldonado, "Sonia: A service for organizing networked information autonomously," in *Proc. Digital Libraries 3rd ACM Conf. Digital Libraries*, 1998, pp. 200–209.

[16] A. El-Hamdouchi and P. Willett, "Hierarchic document classification using ward's clustering method," in *Proc. 9th Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Pisa, Italy, 1986, pp. 149–156.

[17] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *Proc. World Text Mining Conf.*, Boston, MA, 2000.

[18] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, S. Chaudhuri and D. Madigan, Eds., San Diego, CA, 1999, pp. 16–22.

[3] http://wordnet.princeton.edu/

[19] D. Boley, M. Gini, R. Gross, H. Eui-Hong, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore, "Partitioning-based clustering for web document categorization," *Decision Support Syst.*, vol. 27, no. 3, pp. 329–341, 1999.

[20] N. Slonim and N. Tishby, "Document clustering using word clusters via the information bottleneck method," in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, vol. 34, Athens, Greece, 2000, pp. 208–215.

[21] X. Liu, Y. Gong, W. Xu, and S. Zhu, "Document clustering with cluster refinement and model selection capabilities," in *Proc. 25th Int. Conf. Research and Development in Information Retrieval*, Tampere, Finland, 2002, pp. 191–198.

[22] W.-C. Wong and A. Fu, "Incremental document clustering for web page classification," in *Proc. IEEE Int. Conf. on Info. Society in the 21st Century: Emerging Technologies and New Challenges*, Japan, 2000.

[23] H.-J. Kim and S.-G. Lee, "Building topic hierarchy based on fuzzy relations," *Neurocomputing*, vol. 51, pp. 481–486, 2003.

[24] L. Massey, "On the quality of art1 text clustering," *Neural Netw.*, vol. 16, no. 5–6, pp. 771–778, 2003.

[25] R. Kondadadi and R. Kozma, "A modified fuzzy art for soft document clustering," in *Proc. Int. Joint Conf. Neural Networks. IJCNN'02*, vol. 3. Honolulu, HI, 2002, pp. 2545–2549.

[26] T. Kohonen, *Self Organizing Maps Second Edition*, 2nd ed. New York: Springer-Verlag, 1997.

[27] A. Georgakis, C. Kotropoulos, A. Xafopoulos, and I. Pitas, "Marginal median som for document organization and retrieval," *Neural Netw.*, vol. 17, no. 3, pp. 365–377, 2004.

[28] V. J. Hodge and J. Austin, "Hierarchical growing cell structures: Treegcs," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 2, pp. 207–218, 2001.

[29] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1331–1341, Nov. 2002.

[30] H. Chen, C. Schuffels, and R. Orwig, "Internet categorization and search: A self-organizing approach," *J. Vis. Commun. Image Represent.*, vol. 7, no. 1, pp. 88–102, 1996.

[31] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.

[32] O. Zamir, O. Etzioni, O. Madani, and R. M. Karp, "Fast and intuitive clustering of web documents," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, 1997, pp. 287–290.

[33] S. M. Weiss, B. F. White, and C. V. Apte, "Lightweight document clustering," in *Proc. Principles of Data Mining and Knowledge Discovery 4th Eur. Conf.*, Lyon, France, 2000, pp. 665–672.

[34] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[35] H. Yin and N. M. Allinson, "On the distribution and convergence of the feature space in self-organizing maps," *Neural Computat.*, vol. 7, no. 6, pp. 1178–1187, 1995.

[36] E. de Bodt, M. Cottrell, and M. Verleysen, "A statistical tool to assess the realibility of self-organizing maps," in *Advances in Self-Organizing Maps*, N. Allinson, H. Yin, L. Allinson, and J. Slack, Eds. Berlin, Germany: Springer-Verlag, 2001, pp. 7–12.

[37] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.

[38] S. Gunter and H. Bunke, "Validation indexes for graph clustering," *Pattern Recognit. Lett.*, vol. 24, no. 8, pp. 1107–1113, 2003.

[39] A. Rauber, E. Pampalk, and J. Paralic, "Empirical evaluation of clustering algorithms," *Zbornik Radova J. Inf. Org. Sci.*, vol. 24, no. 2, pp. 195–209, 2000.

[40] H. Ye and B. Lo, "Toward a self-structuring software library," *Inst. Elect. Eng. Proc. Software*, vol. 148, no. 2, pp. 45–55, 2001.

[41] D. Pullwitt, "Integrating contextual information to enhance som-based text document clustering," *Neural Netw.*, vol. 15, no. 8–9, pp. 1099–1106, 2002.

[42] P. G. Babu and N. M. Murty, "A near-optimal initial seed value selection in k-means algorithm using a genetic algorithm," *Pattern Recognit. Lett.*, vol. 14, no. 10, pp. 763–769, 1993.

[43] R. Freeman, H. Yin, and N. M. Allinson, "Self-organizing maps for tree view based hierarchical document clustering," in *Proc. World Conf. Computational Intelligence; Int. Joint Conf. Neural Networks*. Honolulu, HI, 2002, pp. 1906–1911.

[44] R. Freeman and H. Yin, "Self-organizing maps for hierarchical tree view document clustering using contextual information," in *Proc. Intelligent Data Engineering and Automated Learning*, vol. 2412, 2002, pp. 123–128.

[45] G. Salton, *Automatic Text Processing—The Transformation, Analysis, and Retrieval of Information by Computer*. Reading, MA: Addison-Wesley, 1989.

[46] A. El-Hamdouchi and P. Willett, "Techniques for the measurement of clustering tendency in document retrieval systems," *J. Inf. Sci. Princ. Pract.*, vol. 13, no. 6, pp. 361–365, 1987.

[47] N. Slonim, N. Friedman, and N. Tishby, "Unsupervised document classification using sequential information maximization," in *Proc. 25th Int. Conf. Research and Development in Information Retrieval*, Tampere, Finland, 2002, pp. 129–136.

[48] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

Richard T. Freeman (S'00) received the M.Eng.degree in computer systems engineering and the Ph.D. degree in content management and neural networks, both from the University of Manchester, Manchester, U.K., in 2000 and 2004, respectively.

His main interests are enterprise portals, enterprise content management, knowledge management, and search engine technology. He is currently working for a large multinational IT consulting and services company on information management related projects.

Hujun Yin (S'92–M'96–SM'03) received the B.Eng. degree in electronic engineering and the M.Sc. degree in signal processing, both from Southeast University, Nanjing, China, in 1983 and 1986, respectively, and the Ph.D. degree in neural networks from the University of York, York, U.K., in 1996.

He worked as an Assistant Lecturer and Lecturer at Tongji University, Shanghai, China, from 1986 to 1991. From 1996 to 1998, he worked as a Postdoctoral Research Associate at the University of Manchester Institute of Science and Technology (UMIST), now merged with the University of Manchester, where he was appointed a Lecturer in 1998 and promoted to Senior Lecturer in 2003. His research interests include neural networks, self-organization in particular, pattern recognition, image processing, bioinformatics, and knowledge management. He has published over 60 papers in leading international journals and international conferences. He has co-organized a number of international conferences (e.g., WSOM'01, IDEAL'02, and IDEAL'04). He served as a guest editor for *Neural Networks*, *Journal of Mathematical Modeling and Algorithms*, and the *International Journal of Neural Systems*.