Map-based Interfaces for Information Management in Large Text Collections

Rudolf Mayer, Angela Roiger, Andreas Rauber
Institute of Software Technology and Interactive Systems
Vienna University of Technology, Vienna, Austria
mayer@ifs.tuwien.ac.at, angela@roiger.at, rauber@ifs.tuwien.ac.at

Journal of Digital Information Management

***ABSTRACT**:The Self-Organising Map (SOM) has been proposed as an alternative interface for exploring Digital Libraries or other big document collections, in addition to conventional search and browsing. With advanced visualisations assisting the user in understanding the contents of the map and its structure, as well as advanced interaction modes as zooming, panning and area selection, the SOM becomes a feasible alternative to classical search interfaces. Several applications show the SOM's utility for this task. However, there are still shortcomings in helping the user understanding the ma, which is essential to fully exploit the SOM's potential as an Information Management tool. There are insufficient methods developed for describing the map to support the user in the analysis of the map contents. In this paper, we give an overview of existing techniques and applications of SOMs in Digital Libraries, and present recent work in assisting the user in exploring the map by automatically describing maps using advanced labelling and summarisation of map regions, focusing on text collections. Therewith, the SOM becomes an attractive tool for Information Management in large corpora.*

Categories and Subject Descriptors
H.3.7 [Digital Libraries]; H.3.5 [Online Information Services]: **I.7 [Document and Text Processing]**

General Terms
Self-organizing maps, Neural network, Information Management

Keywords: Map interfaces, Text collection

Received 10 Sep. 2007; Reviewed and accepted 27 Jan. 2008

1. Introduction

The Self-Organising Map (SOM) [1] is a popular unsupervised neural network model that provides a mapping from a high-dimensional input space (for example text documents described in a vector space model) to a low, often two-dimensional, output space. The mapping of the SOM is topology preserving – elements close in the input space will in general also be close in the output space. Due to its interesting properties, the SOM has been used in many data mining settings, for example in several applications to automatically organise document collections in a Digital Library by their content. Examples for such collections are in the domain of text documents, as in the SOMLib Digital Library system [2] or in a map of news texts [3], music documents as in the SOMeJB system [4], or images as in the PicSOM system [5]. As a recent example, also the Digital Library Management System (DLMS) developed by the DELOS Network of Excellence [6] incorporates the SOM as an interface to a Digital Library's content, as it offers the user support in analysing and exploring the content. With advanced visualisations and interaction possibilities, the user can exploit the full potential of the SOM. However, we still lack techniques to adequately help the user in analysing the contents of the map. For large maps, containing several tens of thousands documents describing various different topics, it becomes increasingly difficult to quickly understand the map.

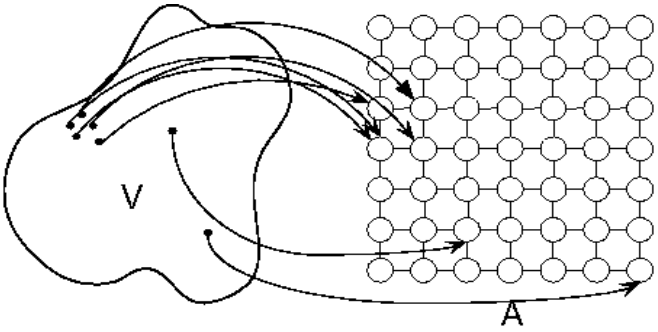

Figure 1. Mapping of the SOM: Spatially close elements in the input space *V* are spatially close in the output space *A* as well

In this paper, we give an overview of existing applications of the Self-Organising Map in Digital Libraries and techniquesto explore and interact with the map. Furthermore, we present recent work in making the SOM more usable for Information Management by automatically describing regions in the map through adding semantic labels to the SOM, using clustering methods to identify topical areas and selecting representative labels for those regions. Moreover, we present work on automatically summarising the content of those regions on the SOM.

The remainder of this paper is organised as follows: Section II gives a brief overview of the Self-Organising Map and its application in the context of Digital Libraries. Section III describes our work in labelling and summarising regions, while Section IV presents the experiments conducted. Section V presents conclusions and future work.

2. Self-Organising Map

The Self-Organising Map (SOM) is a neural network model frequently employed for various data mining purposes. It provides a mapping from a high-dimensional input space to a lower-dimensional output space. Although many different architectures exist, the output space is in many applications organised as a two-dimensional rectangular grid of units, a

representation that is easily understandable for users due to its analogy to conventional two-dimensional maps. An illustration of this mapping is given in Figure 1.

Each of the units on the map is assigned a weight vector, which is of the same dimensionality as the vectors in the input space. During the SOM training algorithm, these these weight vectors are adjusted by a sort of competitive learning in a way that the map best describes the domain of observations. The training process consists of the following basic steps:

- Initialisation of the map
- A number of iterations of
  - Presenting input patterns and finding the best matching unit
  - Adapting the weight vectors of the best matching unit and a certain number of neighbouring units.
- Fine-tuning

For initialisation, a basic and common approach is to initialise each unit with a randomly generated weight vector. Alternatively, randomly chosen vectors from the input data set could be taken as initial values. More sophisticated methods include principal component analysis, where the first two eigenvectors are used for intialising the map.

After initialisation, the actual learning phase is carried out. The number of learning steps (iterations) depends mainly on the number of items in the input space – a common approach is thus to set the number of iterations as a multiple of the number of input items. In each training iteration, a vector of the collection of input patterns is randomly selected. It is then presented to the SOM, and the unit (i.e. the unit's weight vector) which is most similar to a presented input vector x, referred to as the winner or best matching unit (BMU) c, is selected according to:

$$c(x;\, t) = arg \min_i \{d(x(t), m_i(t))\} \quad (1)$$

As a measure $d(x;mi)$ for the distance between the weight vector mi and the input vector x several different metrics have been studied. Probably most common, the L2 norm or Euclidian distance is employed. Other commonly used distance metrics are the L1 norm (also called city-block or manhatten distance), and the cosine distance.

The SOM then is learning from the input sample to improve the mapping quality, i.e. some weight vectors of the SOM are adapted towards x. The new values of the weight vectors are determined by their current value and two other factors, the learning rate as well as the neighbourhood function hci as follows:

$$m_i(t+1) = m_i(t) + \alpha(t)\, h_{ci}(t)[x(t) - m_i(t)] \quad (2)$$

The learning rate α, $0 < \alpha(t) < 1$, determines how much a vector is adapted, and is normally a time-decreasing function, i.e. vectors are adapted more in the beginning of the learning process, with this adaptation decreasing towards the end.

The neighbourhood function is typically symmetric around the winning unit. A simple approach is to define a neighbourhood set N_c around the best matching unit c. Nodes which lie inside this neighbourhood set are, all to the same degree, adapted according to the learning rate, while units outside the neighbourhood set are left as they are. A more sophisticated and propably more widely used approach is the use of a Gaussian function, which adapts the units' weight vectorsdifferently depending on their 'distance' from the winning unit. With r_i and r_c denoting the coordinates of the units i and c in the two-dimensional output space $\mathbb{R}^2$, respectively, it can be given as

$$h_{ci}(t) = e^{\frac{||r_i - r_c||^2}{2.\sigma(t)}} \quad (3)$$

The Gaussian function generally is chosen to be decreasing with time t by the usage of a monotonically decreasing function $\sigma(t)$.

In the fine tuning phase, all input vectors are once more presented to the to the map, but in this step only the winner is adapted, with a very low learning rate. The rational behind is that at this stage, the global ordering of the map is already achieved.

An important property of the SOM is that the mapping is (to a high extend) topology preserving – elements which are located close to each other in the input space will in general also be closely located in the output space, while dissimilar patterns will be mapped on opposite regions of the map. This is illustrated in Figure 1, where the 'cloud' of objects located in the same area in the input space ***V*** gets mapped on four neighbouring units in the output (map) space *A*.

The SOM thus provides a sort of clustering of the data, however, without explicitly assigning data items to clusters. It does not identify cluster boundaries as opposed to, e.g, the k-Means method.

The SOM algorithm can in general not perfectly preserve the topology while mapping from a high to a low dimensional space, thus topology violations may occur. This becomes apparent e.g. when for a given input vector a second-best matching unit would be further away from the best-matching unit than a third-best matching would be. Several different quality measures have been developed to detect such errors.

It is also worth noting that the x and y coordinates of the SOM have no specific meaning, except to measure the distance between units. Especially, they do not represent any of the features of the input space.

The generated map can help the user in getting a quick overview of the patterns in the input space. With fitting visualisations highlighting boundaries, it also allows an easier interpretation of the clusters and correlations in the content.

A. Self-Organising Maps in Digital Libraries

In the context of Digital Libraries, the input space is mostly a vector-space model representation of the documents the Digital Library holds, which can be in the form of text, images, audio, video, or any other media that can be represented in vectorial form.

Using the the Self-Organising Map as an interface to digital document collections has already been proposed in the WEBSOM project [7], where the contents of a newsgroup collection containing a million of articles was clustered on the map. The application provides the user with a map of the document collection which she can zoom into by predefinedlevels and navigate in. On the most detailed zooming level, a list of the documents mapped onto that region of the map is provided.To add semantic meaning, the map gets automatically labelled by the names of the most dominant corresponding newsgroups, which is a feasible approach when some kind of categorisation is available for the documents.

The SOM as an interface to Digital Libraries has further been demonstrated in the SOMLib Digital Library System [2]. The SOMLib system utilises the SOM and other techniques to

support the user by employing as many concepts as possible which she already knows from a conventional library. Similar to a map of a conventional library, depicting the arrangement of the shelves where books on certain topics are located, the SOM gives an overview of the contents of the Digital Library. Similar to finding related books in the same shelf, once the user has found a specific document of interest, she can find documents that are related in content in the neighbourhood of that document. A symbolic visualisation of bookshelves using the LIBViewer [8] method further supports this metaphor. The SOMLib system utilises a labelling algorithm called LabelSOM [9] to automatically add semantic descriptors to the single map units, without having the need for any category information being available. This method is described in more detail in Section III-B, where we will also present an extended version.

In [3], the SOM is used to create a web-based knowledge map of news articles. The application supports hierarchical zooming into the map in predefined levels of zoom. Besides the map, also a hierarchical list of topics is displayed as an alternative for users who prefer a one-dimensional visualisation. Skupin [10] uses clustering on the map to apply labels to regions, which are generated based on term and document frequencies, using a *tf X idf*-based approach. The focus in that work is on the visualisation, which tries to resemble geographical maps as closely as possible. This is achieved using a separate GIS software system. The interaction possibilities for the user are limited - zooming is in predefined levels, the labels cannot be changed interactively, and only one type of visualisation is available.

Based on the concept of the SOMLib system, a sophisticated desktop client software has been developed, which allows for various ways of user interaction with the Digital Library content [11]. With stepless zooming and panning functionalities, the user can analyse the content at any desired detail level, from viewing the whole map at once down to viewing single documents. Tools for selecting rectangular regions or units along a path allow the user to select several documents at once, and open them for example in a text viewer or in an audio application for playlist generation. One important aspect of SOMs as interfaces to Digital Libraries is that they should not be meant to replace traditional query and retrieval techniques, but rather be complementary to it. This approach is presented in [12], where a Self-Organising Map is integrated into the popular open-source Digital Library system Greenstone [13] as anew service, based on the existing query services such as search or list browsing. That way, the user can still use all the basic functionalities provided by Greenstone, but she will also be able to use the additional information the SOM mapping provides about the documents matching the query results and the whole collection itself. A screenshot of the system is depicted in Figure 2, where the topleft side figures the traditional Greenstone search interface, the lower part the document result listing, and the right part holds the map.

The map can be used in two different ways. First, results of the Greenstone search will be highlighted on the map by markers. The user can immediately see which documents have a topical similarity, as these documents will usually all be located close to each other and form a cluster on the map. This way, distinguishing between different topics found on an ambiguous search term as e.g. 'jaguar' becomes easily – the documents referring to the sports car will be clearly separated from those talking about the animal because they appear in a different context together with other words. Additionally, outliers found via the search become visible as isolated spots on the map. Secondly, the user can explore the map – she can select units, upon which the documents lying on that units will be added to the result list of the Greenstone search. This allows the user to retrieve potentially relevant documents for a specific information need, even if they were not initially retrieved by the (usually rather short) query issued. Documents that have been matched both by the map selection and the search result will be marked especially, as they may be of higher importance. The user can get additional information on the content of the collection by mouse-over popups displaying terms that best describe the documents on a certain unit. A small user study suggested that this interface is suitable for Digital Libraries once the user has become a bit familiar with the map.

The SOM has also been used for other types of media besides text. In the SOMeJB project [4], the concept of the SOMLib system has been extended to audio and music documents. documents. Similar to the SOMLib system, the SOMeJB arranges musical pieces, described by a set of feature vectors extracted solely from the audio content, into topical clusters by the sound characteristics, as the user is familiar with from a traditional record store. In the PicSOM project [5], the SOM has been used for Information Retrieval in image databases, incorporating methods of relevance feedback.

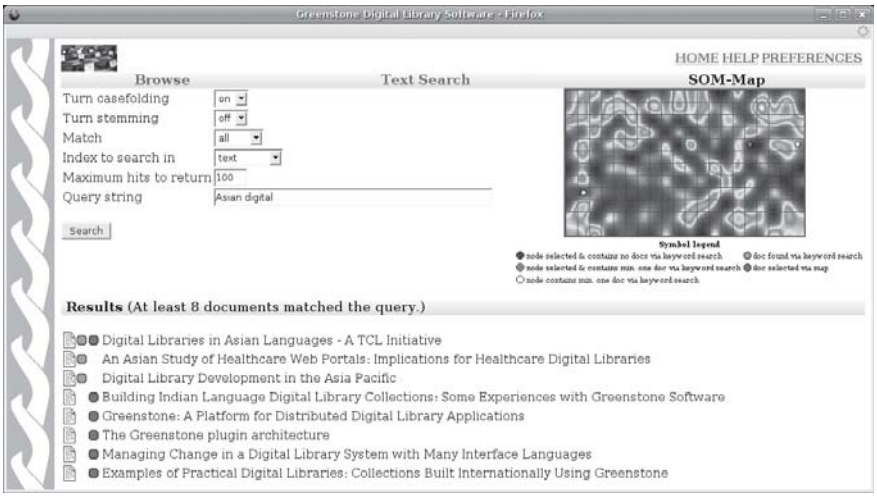

Figure 2. Enhancing the traditional Greenstone query search with a SOM map

3. Describing the Self-Organising Map Regions

In this section we present our work on identifying and describing regions in the mapping generated by the Self-Organising Map. As the SOM does not generate a partition of the map into separate clusters, we utilise another clustering algorithm on the weight vectors of the units to identify the regions (Section III-A). Applying the LabelSOM method (Section III-B), we create semantic labels for those regions (Section III-C) that assist the user in getting a first glance overview of the contents of the map. To further support the analysis phase, we additionally provide summarisation of documents of the regions using Automatic Text Summarisation methods (Section III-D).

A. Clustering

Clustering is an unsupervised machine learning process of finding natural groupings amongst unlabelled objects. The members of a cluster are similar to each other in some characteristics, and are dissimilar to members of other clusters.

Although the SOM already groups similar objects next to each other, it does not generate an explicit partitioning into separate groups. Thus, to identify topical regions, we are clustering the units of a SOM by clustering the weight vectors of the SOM units.

We are using an agglomerative hierarchical clustering algorithm. At the start of an agglomerative clustering process, every unit lies in its own cluster. In each subsequent step, bigger and bigger clusters are built by grouping similar clusters together, until finally only one cluster, containing the whole dataset, remains. This type of algorithm is inverse to divisive hierarchical clustering, which starts from one big cluster that is broken into smaller clusters on each step.

Specifically, we use Ward's linkage [14], also known as minimum variance clustering, as one of the most performant within the linkage clustering families. In this algorithm, the similarity of two clusters is measured by the distance of each pair of clusters, and is defined by the increase in the 'error sum of squares' *ESS*, if the two clusters are to be combined. The ESS of a cluster of $|X|$ values is defined as:

$$ESS(X) = \sum_{i=1}^{|X|} |x_i - \frac{1}{|X|} \sum_{j=1}^{|X|} x_j|^2 \quad (4)$$

Subsequently, the distance D between two clusters X and Y can be defined as:

$$D(X; Y) = ESS(XY) - [ESS(X) + ESS(Y)] \quad (5)$$

where XY is the union of clusters X and Y.

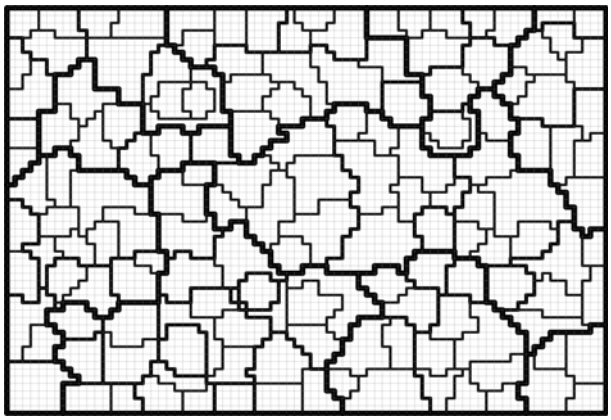

Figure 3. Three different layers of clusters on the SOM

The result of the Ward's algorithm is a hierarchy of clusters which the user then can 'browse' through, to support the analysis of the clusters in the data beyond the insight the SOM alone can give. Increasing the number of displayed clusters means splitting existing clusters into two new ones, while reducing the number of clusters is achieved by merging two clusters into one. This is advantageous over a non-hierarchical clustering algorithm, such as for example K-means, where changing the number of clusters might heavily change the size and layout of clusters. This is obviously not a desired behaviour when we want to allow the user to interactively analyse the map contents by means of changing the number of clusters. Moreover, hierarchical clustering allows us to display multiple layers of clusterings at the same time, to illustrate how the data is structured coarsely and in detail. This is feasible as, in contrast to non-hierarchical clustering algorithms, the clusters of a layer with more clusters can never be 'cut' by clusters of a layer with fewer clusters, and thus not disrupt the illustration. This is illustrated in Figure 3, where we have clustered the SOM in three different layers of nine, 50, and 150 clusters, represented by cluster boundaries in different line strenghts.

B. Labelling units with LabelSOM

To assist the user in semantically interpreting the regions of the SOM, we automatically generate labels for the clusters. The cluster labels are based on the units labels generated by the LabelSOM method [9] which assigns labels to the units of the SOM describing the features of the data points mapped onto the respective unit, by analysing their mean and variance. This is done by utilising the *quantisation error* qi_k of the vector elements, which is defined for each feature as the sum of the distances between the unit's weight vector mi and all the input vectors $x_j \in c_i$, i.e. the vectors mapped onto the unit i, formally given as:

$$qi_k = \sum_{x_j \in c_i} \sqrt{(mi_k - xj_k)^2} \quad k = 1 \ldots .. n \quad (6)$$

This means that a low quantisation error characterises a feature that is similar in all input vectors to the weight vector. Thus the assumption is made that such a feature describes the objects mapped on this unit well. If however the input vectors mapped on one specific unit all have in common that some features are not representative and therefore have the value of 0, those features often also have a quantisation error of almost 0 for a unit. Such features, though. are in most application domains not appropriate for labelling the unit, since this would describe what the unit does not contain, rather than what it contains. Therefore, we may require a feature to also have a minimum average value, calculated from all the input vectors mapped to the unit.

Note that the LabelSOM method relies on having meaningful names for each feature. Alternatively, also class labels (manually) assigned to the input vectors could be utilised as labels.

C. Labelling Regions

To choose a label for a region, we consider only the unit labels present in that cluster, as the unit labels are already a selection of features describing the contents of each unit. This method is thus faster in computation than checking all possible features.

Depending on the data and application, it is preferable to choose the region label based upon a low average quantisation error, a high mean value, or a combination of

both. Therefore we offer the user the possibility to interactively assign priority weights for those two measures, to achieve more meaningful labels. Making use of the properties of the hierarchical clustering as described in Section III-A, we can also display two or more different levels of labels, some being more global, some being more local.

In the visualisation of the SOM, the labels are placed in the centroid of the cluster, which may result in some overlapping labels, especially if more layers of cluster are displayed at the same time. To achieve a clear arrangement, the labels can be manually moved on the map, or adjusted in their size and rotation. For some labels, it might also be useful to edit their text, for example if the label text is only a word stem as in our experiment described later in this paper, or if the labels on the highest layer of clusters should be described rather by highlevel categories, rather than terms derived from the content only.

D. Region Summarisation

Even though labelling the map regions may assist the user in quickly getting a coarse overview of the topics, labels can still be ambiguous or not convey enough information. Therefore, we also employ methods from the Automatic Text Summarisation domain. Based on the regions identified from the clustering process, we automatically provide a short summary of the contents of the documents mapped onto those regions, allowing the user to get a deeper insight into the contents.

Automatic text summarisation [15] tries to automatically generate a summary of one or more texts to present the main ideas of the contents in a short and compact form. Different approaches and algorithms in the field can in principle be distinguished on the range of documents they operate on (single document vs. multi-document summarisation), and the form of summarisation (extraction vs. abstraction).

Single document summarisation deals with providing a summary of a single document, while multi-document summarisation, on the other hand, deals with generating summaries of a whole collection of documents. Simple approaches of multi-document summarisation would just treat each single document separately, generate the summary of it, and then present all the summaries to the user. This approach of course does not consider redundancy in the summaries, and does not favour passages that are relevant for a higher number of documents. More advanced techniques would thus treat the whole document collection at once, and present the sentences which are most important concerning all documents. Redundant sentences are eliminated first by applying a measure for overlapping words, and removing sentences with too high similarity.

Further, automatic text summarisation methods can be distinguished by the way they generate the summaries. Extraction as the in most cases easier method tries to identify the most relevant sentences in a document, and presents them as a summary. Abstraction as the more sophisticated approach tries to generate an abstract of the document, which corresponds more to the human understanding of summaries.

Although we also provide summaries of single documents in our application, the main focus is to assist the user in analysing the contents of the Digital Library by providing summaries of the previously identified regions using multidocument summarisation. The application allows the user to select whole regions, or manually any other rectangular shape or units a long a path. For the chosen documents, the user can choose from several different summarisation algorithms using different weighting schemes to determine the importance of sentences for the summaries, and can also specify the desired length of the summary, measured in percent of the original sentences.

4. Experiments

The following experiments were performed using the 20 newsgroups data set[1], a big benchmark corpus which has become very popular for text experiments in the field of machine learning. The data set consists of newsgroup postings from the 20 newsgroups listed in Table I. Each newsgroup contains approximately 1,000 articles from the year 1993.

Several different versions of the dataset exist, we chose one where duplicate messages (i.e. mostly cross-postings

alt.atheism	rec.sport.hockey
comp.graphics	sci.crypt
comp.os.ms-windows.misc	sci.electronics
comp.sys.ibm.pc.hardware	sci.med
comp.sys.mac.hardware	sci.space
comp.windows.x	soc.religion.christian
misc.forsale	talk.politics.guns
rec.autos	talk.politics.mideast
rec.motorcycles	talk.politics.misc
rec.sport.baseball	talk.religion.misc

Table 1. The 20 Newsgroup Data Set

over two or more different newsgroups) were removed, resulting in a total of 18,828 documents, and where each posting consists of the message body and in addition the 'Subject' and the 'From' header lines, but other header lines have been removed.

In our experiments we used a standard bag-of-words indexing approach, i.e. our collection of documents is represented as an (unordered) collection of all the words occurring, disregarding grammar rules and word orders. Porter's stemming algorithm [16] was applied to remove prefixes and suffixes to obtain word stems. From the remaining word stems, the features for the input vectors were selected according to their document frequency, and the weights are computed using a standard *tf* X *idf* weighting scheme [17]. This resulted in a 3,151 dimensional feature vector for each document, from which several maps were trained. The specific map we will use in the remainder of this paper to illustrate our results is of 75x55 units in size. Thus, each SOM unit will in average represent 4.5 documents.

A. Cluster Hierarchy Browsing and Labelling Regions

In our application it is possible to explore the clustered SOM interactively: to view the different levels of clustering and to zoom into the map to view the single postings. The user can browse the clustering levels either viewing only the cluster borders, or also highlighting each cluster in a different colour. The former is shown in Figure 4, illustrating the steps Fig. 5. Nine coloured and labelled first-layer clusters, with 67 smaller secondlayer clusters with labels from one to nine clusters. Note that the label 'window' stands here for 'windows'. There is a special cluster in the lower right-handcorner with the a label also used on other clusters – 'god' in

[1] http://people.csail.mit.edu/jrennie/20Newsgroups

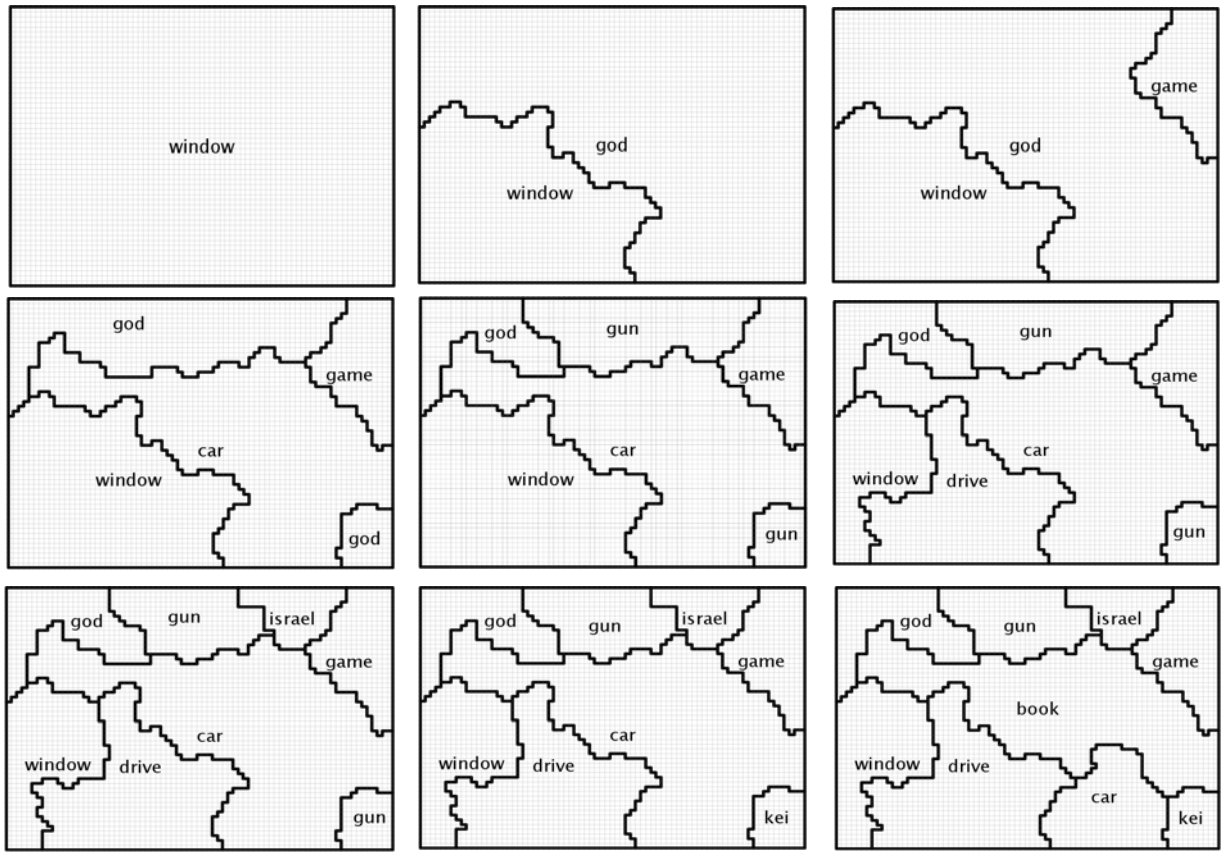

Figure 4. 1 – 9 Clusters with automatically generated labels

the fourth step, and 'gun' in the steps five to seven. This cluster is, however, not a separate one - when viewing the clustering with colours,it becomes apparent that this area is part of the disjoint clusters 'god' and 'gun', respectively, in the upper part of the map. Such disjoint clusters may stem from topologyviolations during the mapping progress. With the clustering applied on top of the trained map, such topology violations can not only become apparent to the user in a fast and easy way, but also indicate which related objects might have been erroneously mapped to different regions. With this interactive exploration of the clusters, the user can thus gain valuable information about the structure of the documents in the collection.

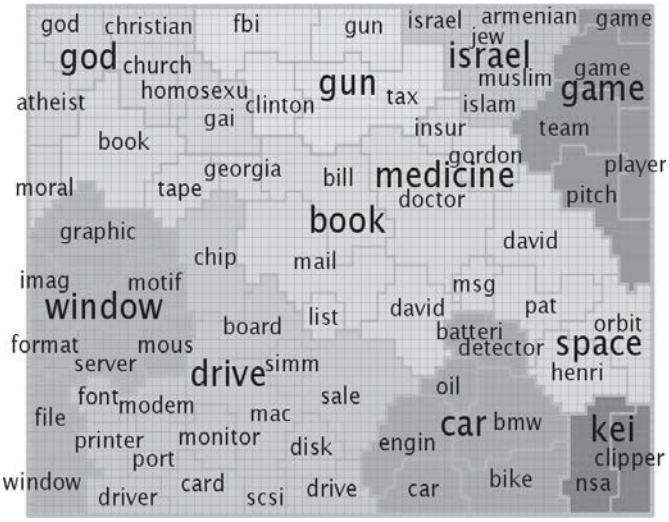

Figure 5. Nine coloured and labelled first-layer clusters, with 67 smaller second-layer clusters with labels

Figure 5 shows nine clusters with larger labels, while Figure 6 shows the same nine clusters on the first layer, and in addition a refinement of the clusters on a second layer containing 67 clusters, with smaller labels. The two clusters labelled 'david' are in fact one disjointed cluster. As a result of the stemming algorithm, words ending with an *y* now end with an *i*, for example the labels 'kei' (containing most of sci.crypt) or 'batteri' (various postings e.g. from sci.electronics or rec.motorcycles). There are also some labels where obviously the suffixes of the original words have been removed, as in the labels 'imag' or 'insur'. The labels were not edited to show the terms based on which the map has been created.

In the top right-hand corner is a large cluster labelled 'game' containing most postings from the two sports related newsgroups. The large cluster next to it labelled 'israel' contains mainly postings from talk.politics.mideast. In the upper left-hand corner there is a cluster labelled 'god' containing all the newsgroups dealing with religion, i.e. alt.atheism, soc.religion.christian and talk.religion.misc. It is interesting to note that, in contrast to the newsgroup hierarchy, where these groups lie in three different top level hierarchies, they are located in the same area on the Self-Organising Map, and are combined into one cluster.

The large cluster in the middle labelled 'book' contains many small clusters of which only a few have meaningful labels. The small clusters labelled 'insur' and 'doctor' suggest that they contain postings from the sci.med newsgroup and the cluster label 'orbit' relates to the newsgroup sci.space. The labels containing names such as 'gordon', 'david' or 'bill' do not help in identifying the underlying topics in those areas of the map. However, names cannot be easily automatically removed during the document indexing process, as some common names as Mark or Bill are also verbs nouns,

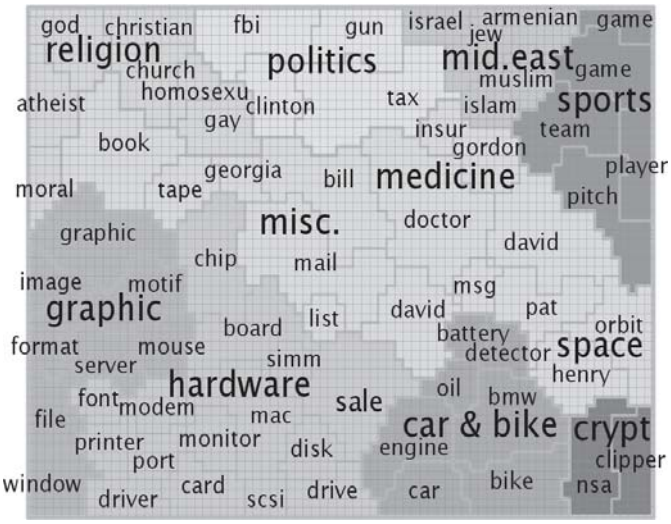

Figure 6. Nine coloured first-layer clusters, with 67 smaller second-layer clusters, labels manually edited

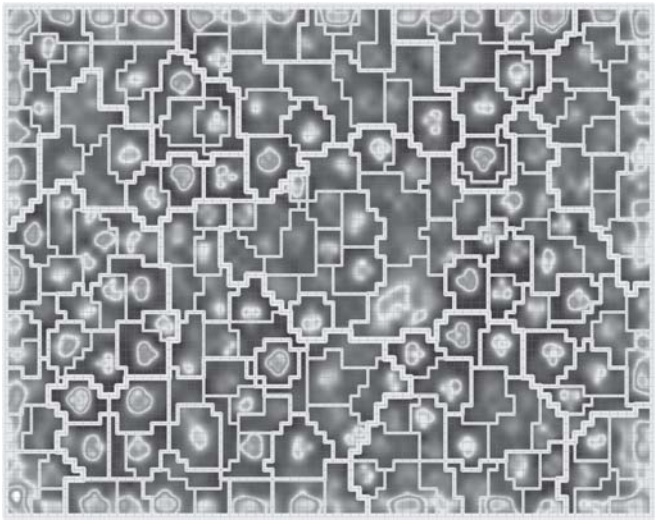

Figure 7. Comparison of cluster boundaries from Ward's linkage clustering and Smoothed Data Histograms Visualisation

respectively. Furthermore, names can sometimes be useful as labels, for example if they refer to a famous person. In the specific map of our experiments, they however mainly refer to the names of frequent posters, and can thus also give indications on areas where the discussion is dominated by some specific people.

The small cluster labelled 'drive' lies in the cluster with the hardware topics but also directly next to the cluster labelled 'car'. It implies that in this area lies a transition of the word *drive* being used in the meaning of *hard disk drive* to the meaning of *to drive a car*.

To enhance the comprehensibility of the map some labels are manually edited, which is shown in Figure 6: word endings have been added and the labels of the larger areas have been edited to better suit the diverse topics. For example the cluster automatically labelled 'car' is extended to 'car & bike' to point out both newsgroups contained in this cluster. The cluster previously labelled 'gun' containing the newsgroups talk.politics.guns and parts of talk.politics.misc and talk.politics.mideast is adapted to 'politics'. The large cluster in the middle is manually described with three labels to point out the various topics inside.

To give a comparison of the boundaries identified by the Ward's linkage clustering algorithm we contrast the clusters with a visualisation aimed at illustrating structures and cluster boundaries on a SOM. For this purpose, we utilise the Smoothed Data Histograms (SDH) [18] visualisation. The SDH method is based on the idea that clusters are areas in the input space with high densities, and the SDH thus is a visualisation technique displaying an estimation of the probability density of the data on the map. It differs from a simple hit-histogram by taking into account not only the best-matching unit of the input vectors, but rather the s best matching units, adjusted by a weight for their distance to the best-matching unit. Figure 7 shows the same map as before, applying the SDH method. Using a fitting colour-palette, the resulting illustration depicts clusters e.g. as 'islands', and boundaries as 'water', as it is the case in Figure 7. On top of the SDH, three layers of clusters with nine, 67 and and 190 clusters are depicted. The top-layer boundaries are highly correlating with the boundaries identified by the SDH, illustrated by the dark-blue areas between some islands. Also the next layers of clusters are fitting well with the arrangement of 'water' and 'valleys' dividing the islands on a more detailed level.

As the 20 newsgroups data set also contains class labels for each document, namely the name of the newsgroup it was posted to, the clustering of the map can also be compared to these class labels. We employ a visualisation technique that smoothly colours a SOM according to the distribution and location of the class labels of the input data [19] by utilising graph-based methods such as Voronoi regions, and some heuristic rules to optimise the representation. Figure 8 shows this visualisation, where the different colours on the map represent the different newsgroups, and the light-gray lines indicate the boundaries of the 67 clusters stemming from Ward's linkage algorithm. It can be observered quickly by visual inspection that the cluster boundaries and the edges of classes are identical or very close to each other in most areas of the map. This indicates that both the SOM mapping and the clustering applied afterwards can reveal the structure in the data very well.

The map thus created, visualised and enriched with semantically meaningful descriptors can now be used to interactively present the contained information of the Digital Library in an intuitive way and allow the user to gain more insight on the data collection.

B. Region Summarisation

Figure 9 shows the summarisation of one of the regions in the map, namely the cluster labelled 'oil'. The lower-left part of the interface shows the summarisation module, which allows the user to select a summarisation method, and the desired length of the summary. In the example depicted in the figure, we use a multi-document summarisation, by extracting sentences considering their importance for the whole collection of documents selected. We chose 3% of the selected documents as desired summarisation length. Inspecting the summary of the documents, it becomes quickly clear that the major, dominating topic is about oil in the context of engines of bikes and cars. From the label 'oil' alone, this is not clear, as the term could also suggest topics such as the oil price, the oil crisis, or oil refining methods. Thus, the automatic summaries can become very useful in disambiguating certain label terms. This presumption is also

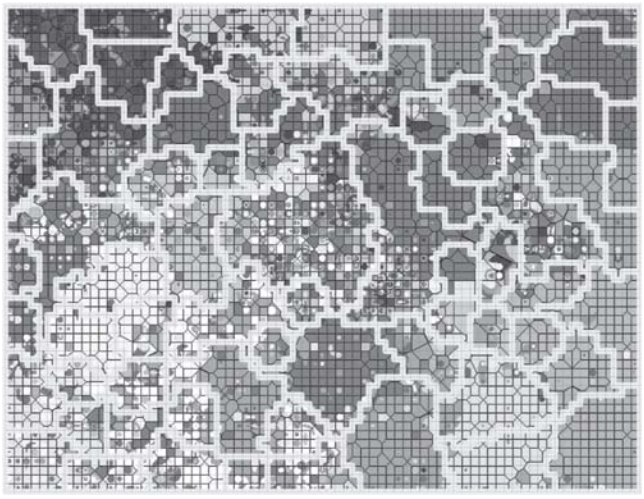

Figure 8. Class labels of the input data and boundaries from Ward's linkage clustering

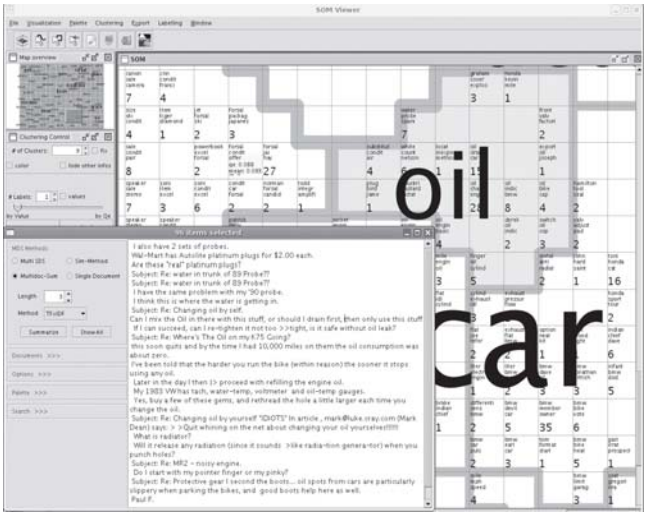

Figure 9. Automatic Summary of the cluster 'oil'

backed by a small user study on the text summarisation, which showed that users find the summaries acceptably comprehensible and useful, and that generally a summary of regions can help in understanding the map better.

5. Conclusion

In this paper we presented the usage of the Self-Organising Map as an interface to Digital Libraries. On top of this well-known approach, we presented recent work on methods to assist the user in interacting with the map. We employ clustering of the SOM to reveal hierarchical structures which can be explored by the user to get a rough overview of the structure of the data on the map. The clustering identifies regions, which we describe on the one hand very concisely by single descriptive words extracted from the document contents, and secondly by applying automatic text summarisation techniques to generate executive summaries of the contents. All methods are integrated into a single application, that provides additional features such as visualisations and advanced interaction via zooming and panning, and selection of arbitrary regions of the map.

With these tools available, the user can be greatly assisted in analysing the SOM generated from the contents of the Digital Library, and therefore getting a quick overview of the contents of the Digital Library itself, and the structure and relationship of the documents it contains, even if the number of documents is huge and their topics diverse. Therewith, the SOM becomes an attractive tool to support Information Management.

References

[1] Kohonen, T. (1995). Self-Organizing Maps, ser. Springer Series in Information Sciences. Berlin, Heidelberg: Springer, Vol. 30.

[2] Rauber, A., Merkl, D. (1999), The SOMLib digital library system, *In*: European Conference on Digital Libraries (ECDL 1999). Paris, France: Springer, September 22-24, p. 323–342.

[3] Ong,T.-H., Chen, H., Sung, Wand., Zhu, B (2005).Newsmap: a knowledge map for online news, *Decision Support Systems*, 39 (4) 583–597.

[4] Rauber, A., Pampalk, E., Merkl, D (2003).The SOM-enhanced JukeBox: Organization and visualization of music collections based on perceptual models, *Journal of New Music Research,* 2003.

[5] Laaksonen, J., Koskela, M., Laakso, S., Oja, E (2000). PicSOM-contentbased image retrieval with self-organizing maps, *Pattern Recogn. Lett.*, 21 (13-14) 1199–1207.

[6] Agosti, M., Berretti, S., Brettlecker, G., del Bimbo, A., Ferro, N., Fuhr, N., Keim, D., Klas,C.-P., Lidy, T. . Norrie, M Ranaldi, P. Rauber, A., Schek, H.-J., Schreck, T.. Schuldt, H., Signer, B., Springmann, M (2007). Delos- DLMS - the integrated DELOS digital library management system," in Digital Libraries: Research and Development, First International DELOS Conference, Revised Selected Papers, ser. Lecture Notes in Computer Science, V. 4877. Pisa, Italy: Springer, February 13-14.

[7] Kohonen, T., Kaski, S., Lagus, K., Salojrvi, J., Paatero, V., Saarela, A (2000). Organization of a massive document collection, I*EEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery,* 11 (3) 574–585.

[8] Rauber, A., M¨uller-K¨ogler, A. (2001). Integrating automatic genre analysis into digital libraries, *In*: Proceedings of the First ACM-IEEE Joint Conference on Digital Libraries, Roanoke, VA, June 24-28 2001, p. 1–10.

[9] Rauber, A., Merkl, D (2001). Automatic labeling of Self-Organizing Maps for Information Retrieval, J*ournal of Systems Research and Inf. Systems (JSRIS),* 10 (10) 23–45.

[10] Skupin, A. (2002). A cartographic approach to visualizing conference abstracts, *IEEE Computer Graphics and Applications*, 22 (1) 50–58.

[11] Neumayer, R., Dittenbach,M., Rauber, A (2005). PlaySOM and PocketSOMPlayer: Alternative interfaces to large music collections, *In*: Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005), London, UK, September 11-15 2005, p. 618–623.

[12] Mayer, R., Rauber, A (2006). Adding SOMLib capabilities to the Greenstone Digital LibrarySystem, *In*: Proceedings of the 9th International Conference on Asian Digital Libraries (ICADL). Springer, November 27-30 2006, p. 486–489.

[13] Witten, I. H. Boddie, S. J., Bainbridge, D., McNab, R. J (2000). Greenstone: A comprehensive open-source digital library software system, *In*: Proceedings of the 5th ACM conference on Digital Libraries. San Antonio, Texas, United States: ACM, 2000, p. 113–121.

[14] Ward, J (1963). Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association*, 58 (301) 236–244.

[15] Mani, I., Maybury, M. T (1999). Advances in Automatic Text Summarization. Cambridge, MA, USA: MIT Press.

[16] Porter, M (1980). An algorithm for suffix stripping, *Program* 14 (3) 130–137.

[17] Salton, G (1989). Automatic text processing – The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Longman Publishing Co., Inc., 1989.

[18] Pampalk, E., Rauber, A., Merkl, D (2002). Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps, *In*: Proceedings of the International Conference on Artifical Neural Networks (ICANN'02). Madrid, Spain: Springer, August 27-30 2002, p. 871–876.

[19] Mayer, R., Aziz,T. A., Rauber, A (2007). Visualising class distribution on self-organising maps, *In*: Proceedings of the International Conference on Artificial Neural Networks (ICANN'07), ser. LNCS, J. M. de S´a, L. A. Alexandre, W. Duch, and D. Mandic, Eds., V. 4669. Porto,Portugal: Springer, September 9 - 13 2007, p. 359–368.