

Pre-print: Richard Freeman and Hujun Yin, "Self-Organising Maps for Hierarchical Tree View Document Clustering Using Contextual Information", in [IDEAL'02](#), [LNCS 2412](#), Springer, 12-14 August, 2002, pp. 123-128.

For more details and enhancements please refer to the journal papers listed on <http://www.rfreeman.net/>

# Self-Organising Maps for Hierarchical Tree View Document Clustering Using Contextual Information

Richard Freeman and Hujun Yin

University of Manchester Institute of Science and Technology (UMIST),  
Department of Electrical Engineering and Electronics,  
PO Box 88, Manchester, M60 1QD,  
United Kingdom

[research@rfreeman.net](mailto:research@rfreeman.net)

[h.yin@umist.ac.uk](mailto:h.yin@umist.ac.uk)

<http://www.rfreeman.net>

<http://images.ee.umist.ac.uk/hujun/>

**Abstract.** In this paper we propose an effective method to cluster documents into a dynamically built taxonomy of topics, directly extracted from the documents. We take into account short contextual information within the text corpus, which is weighted by importance and used as input to a set of independently spun growing Self-Organising Maps (SOM). This work shows an increase in precision and labelling quality in the hierarchy of topics, using these indexing units. The use of the tree structure over sets of conventional two-dimensional maps creates topic hierarchies that are easy to browse and understand, in which the documents are stored based on their content similarity.

## 1. Introduction

With the tremendous growth of digital content on corporate intranets, organisation and retrieval is becoming more and more problematic. Manual document sorting is difficult and inefficient in a highly competitive and fast moving e-market where the corporations have to keep the leading edge over their competitors. This is why knowledge and content management has recently created so much interest. In this paper we propose a method that greatly enhances automated document management using document clustering which automatically organises documents into a generated hierarchy of topics. This taxonomy is automatically built based on the contents of the documents and without using any prior knowledge or metadata. In the indexing stage all the text corpus for each document is analysed and important features are extracted and formed into vectors. Document clustering is then performed using a set of one-dimensional, independently spun, growing Self-Organising Maps (SOM).

This paper is organised as follows. Section 2 briefly introduces general document clustering procedures. Section 3 describes the related work, which used the SOM for document clustering. Section 4 then presents the proposed method. Section 5 presents and discusses the experiments and finally section 6 concludes and suggests possible future work.

## 2. Document Clustering Overview

Document clustering is an area in Information Retrieval (IR) and deals with grouping similar documents together without any prior knowledge of their organisation. In the first stage, document indexing involves transforming text or strings into commonly a vector or histogram form. In general not all the indexed terms are of interest or useful, so feature selection is performed. Then the remaining terms are usually weighted according to their statistical importance. Finally similarity computations are performed amongst documents using those terms and a clustering algorithm.

### 2.1. Indexing Methods

The currently most popular method is the Vector Space Model [1], which is claimed to outperform more sophisticated methods, that rely for example on lists of synonyms or tables of term relationships. Single term indexing is one of the most simple and common methods widely used by the IR community. In this method the frequency of each word in each document is recorded in a large documents Vs words matrix. This single term method is also called “bag of words”, since a lot of semantics and meaning is lost when ignoring the neighbouring words, which provide some context. Another indexing method that uses a succession of words in a short context partially solves this problem. A comparison for these two indexing methods has shown that more accurate terms and a superior cluster quality for multiple successive words they call “lexical affinities” [2]. Other approaches to indexing also exist such as n-grams, linguistic terms or full sentences.

### 2.2. Feature Selection and Term Weightings

In most IR systems the indexing phase is then followed by a feature selection and weighting phase. The feature selection involves discarding common words such as “the”, “of” or “and” stored in a stop list, which are very frequent and alone do not convey much meaning. Then a suffix-stripping algorithm such as Porter’s, is used to stem the words to a common reduced form. The remaining words are then typically weighted using the  $tf \times idf$  [1]. This allows the less frequent words to be given more weighting than the more frequent ones. Following Shannon’s information theory the less frequent the word, the more information value it conveys.

In IR, feature selection is a phase where the potential relevant features are selected and the less important or irrelevant ones are discarded. For text analysis there are broadly speaking two methods that use thresholds or term co-occurrence analysis methods. Thresholds are generally used to discard all the words that occur in most documents, as these are less likely to help discriminate between different documents. Also words that do not occur in many documents can be discarded, as these might be too specific. The other indexing methods involve term co-occurrence analysis, such as in Latent Semantic Indexing [3], Word Category Maps (WCM) [4] or random projection [5].

For more details and enhancements please refer to the journal papers listed on <http://www.rfreeman.net/>

### **2.3. Clustering**

Using the terms previously obtained from indexing and feature selection, document clustering can be performed. Many methods from cluster analysis can be applied in IR, such as partitioning methods like k-means or hierarchical methods like competitive-linkage. In this paper we focus on using SOM to perform the document clustering. The two reasons for using SOM rather than other clustering methods are that it is topologically preserving and clustering is performed non-linearly on the given input data sets. The topologically preserving property allows the SOM applied to document clustering, to group similar documents together in a cluster and organise similar clusters close together unlike most other clustering methods.

## **3. Related Work**

The SOM and its variants have widely been used to cluster documents such as large two-dimensional map [5], a hierarchical set of maps [6], a growing map [7], a set of growing hierarchical maps [8] and tree view based hierarchical maps [9]. There are also other variants SOM algorithms that could be used for document clustering such as the Neural Gas [10], iSOM [11] or ViSOM [12].

The SOM has also previously been used with contextual information as indexing units rather than using the single term indexing representation. In WEBSOM a two-step approach was used [4]. The first step called WCM was to use a window of three words to create a cluster of categories. Then a second level called Document Map was used to cluster documents using the WCM terms. Note that the three words were only used to cluster terms and not to perform the document clustering. This approach was later abandoned in favour of random projection, which gave superior results [5]. In another SOM based approach, full sentences were used as indexing units [13]. The results using sentences as indexing units and no stemming were similar to using single term approaches in combination with more complicated pre-processing like stemming. However this approach was described as slow and requiring large memory storage even with suggested enhancements.

## **4. The Proposed Method**

Most of the methods using the SOM for documents clustering use the “bag of words” method to represent documents, where only isolated words are indexed. In the proposed method we are extending the work on SOM for tree view based document clustering [9] to make use of short contextual information.

### **4.1. Document Pre-processing**

The document pre-processing includes the indexing, feature selection and term weighting as follows:

For more details and enhancements please refer to the journal papers listed on <http://www.rfreeman.net/>

1. Discard any words in the stop list of common words (such as pronouns, articles or prepositions) and apply plural stemming to the remaining words.
2. Index the stemmed words using a sliding window over each sentence.
3. Discard words that occur infrequently or too frequently in all documents.
4. Weight the remaining terms using  $tf \times idf$  [1] and bias factors depending on feature type (1 word, 2 words or 3 words see **Table 1**).
5. Normalise vectors to give words equal importance in short and longer documents.

For the feature selection we have chosen to use an upper and lower threshold to discard infrequent and too frequent terms that are most unlikely to help discriminate between documents. The flexibility in implementation allows a combination a successive single terms, dual terms and triple terms. This allows a weighting bias for each of these features as shown in **Table 1**. There are more complicated approaches such as performing two levels of clustering: the first selects the suitable terms and second clusters the documents, using those selected terms such as in [4][5][8][13].

**Table 1.** Preliminary weighting bias for each type of feature

Feature Type	Weighting
1 word	0.25
2 words	0.35
3 words	0.40

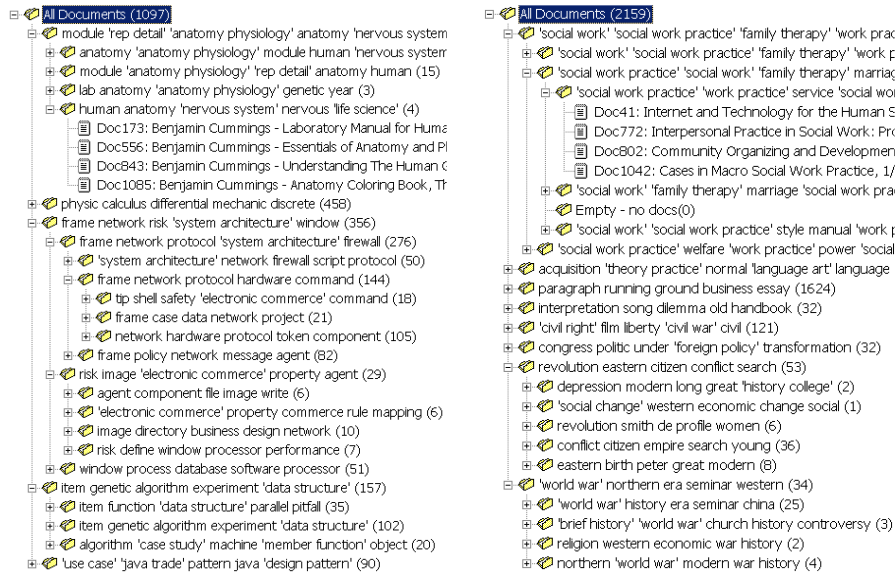
#### 4.2. SOM Procedure

Most previous SOM methods for document analysis used a single or a set of two-dimensional maps [4][5][8][13], which we believe makes realistic large-scale documents browsing problematic. Many existing hierarchical organisation systems use one-dimension to effectively sort the documents, files or books. From the Dewey Decimal Classification (DDC), used worldwide to classify books into a hierarchy of predefined topics, up to the tree view file organisations in Graphical User Interface Operating Systems. So in using one-dimensional SOM, we can directly tie the trained maps as a browsable hierarchical tree, without the need of an extra layer of abstraction as shown in **Fig. 1**. In the Tree View SOM [9] growth is done at map and hierarchical levels. The main idea for these growths is not impose a structure on the documents, but to use the underlying structure to create the taxonomy.

### 5. Experimentation and Discussion

The test data is shown below in **Table 2**. These web documents are variable in both length and content. The content varies greatly from a simple title with one sentence description to table of contents, description of the features, summary, preface and entire articles.

For more details and enhancements please refer to the journal papers listed on <http://www.rfreeman.net/>



**Fig. 1.** Shows two trees of topics generated from *Set B* on the left and *Set C* on the Right, using the proposed method. The labels are chosen by ranking the top five terms with the lowest quantization error at each particular node. Multiple terms are marked with single quotes.

**Table 2.** Test Document Used for Clustering

Set Name	Number of docs	Number of pages	Document source
<i>Set A</i>	618	1 to 16	<a href="http://www.prenhall.com">www.prenhall.com</a>
<i>Set B</i>	1097	1 to 13	<a href="http://www.aw.com">www.aw.com</a>
<i>Set C</i>	2159	1 to 28	<a href="http://www.ablongman.com">www.ablongman.com</a>

*Set A* and *Set B* contain documents on sciences, math and economics but *Set C* deals much more with social issues, politics and literature which are more ambiguous to understand and cluster autonomously. From visual inspection it seems that using multiple words has benefits of accurately indexing terms such as “foreign exchange market”, “molecular biology” or “corporate finance”. However it was also observed that two or three word terms such as “case study”, “companion web site” or “real world” also become part of the corpus, which in some cases may not be desirable.

An advantage of using multiple successive words as indexing units, is that more accurate features such as “document analysis” are taken into account, rather than the isolated words “document” and “analysis”. This creates a basic word disambiguation-indexing scheme applicable to most languages. Another advantage is that this creates a larger set of features from which the most interesting and relevant terms can be selected. Labelling is also made clearer, allowing the user to have a better idea of the contents of each folder or sub-folder without viewing their contents. Visual inspection has shown that the weightings given in **Table 1** provides good labels and clusters for the three data sets. When the weightings were all equal, we observed that many two-word terms naturally ranked within the top five for each cluster, showing their importance to the SOM in the clustering process and their relevance to the underlying topics contained in each cluster.

## 6. Conclusion and Future Work

We have introduced an improved indexing method for the SOM, allowing more precise clustering and more logical labelling. This has been tested on three different realistic data sets of unpredictable content and of greatly variable length. We have also seen that weightings of the short contextual indexing units plays an important role in providing satisfactory results to the overall process of both clustering and labelling. Future work might include looking at the ways to enhance the text processing and feature selection to scale up the number of documents to be clustered. Labelling could also be enhanced with further processing.

## References

1. Salton, G., Automatic text processing: the transformation, analysis, and retrieval of information by Computer, Reading, Mass. Wokingham: Addison-Wesley 1988.
2. Maarek, Y.S., Fagin, R., Ben-Shaul, I.Z., Pelleg, D., Ephemeral document clustering for web applications, IBM Research Report RJ 10186, April, 2000.
3. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R., Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), pp.391-407, 1990.
4. Honkela, T., WEBSOM Self-Organizing Maps of Document Collections, Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6, 1997.
5. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Paatero, V., Saarela, A., Self Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks*, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, vol. 11, n. 3, pp.574-585, May, 2000.
6. Miikkulainen, R., Script recognition with hierarchical feature maps. *Connection Science*, 2(1&2), pp.83-101, 1990.
7. Alahakoon, D., Halgamuge, S.K., Srinivasan, B., Dynamic self organizing maps with controlled growth for knowledge discovery, *IEEE Transactions on Neural Networks*, vol. 11, pp.601-614, 2000.
8. Dittenbach, M., Merkl, D., Rauber, A., The Growing Hierarchical Self-Organizing Map, Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000), vol. 6, pp.15-19, July 24-27, 2000.
9. Freeman, R., Yin, H., Allinson, N. M., Self-Organising Maps for Tree View Based Hierarchical Document Clustering, Proceedings of the International Joint Conference on Neural Networks (IJCNN'02), vol.2, pp.1906-1911, Honolulu, Hawaii, 12-17 May, 2002.
10. Martinetz, T.M., Berkovich, S.G., Schulten, K.J., "Neural-Gas" Network for Vector Quantization and its Application to Time-Series Prediction, *IEEE Transactions on Neural Networks*, Vol. 4, No.4, pp.558-569, July, 1993.
11. Yin, H., Allinson, N.M., Interpolating self-organising maps (iSOM), *Electronics Letters*, Vol. 35, No.19, pp.1649-1650, 1999.
12. Yin, H., ViSOM - A novel method for multivariate data projection and structure visualisation, in *IEEE Transactions on Neural Networks*, Vol.13, No.1, 2002.
13. Pullwitt, D., Der, R., Integrating Contextual Information into Text Document clustering with Self-Organizing Maps, in *Advances in Self-Organising Maps*, Springer, pp.54-60, 2001.