Working Paper Series
ISSN 1170-487X

# Hierarchical Document Clustering Using Automatically Extracted Keyphrases

**By Steve Jones and Malika Mahoui**

Working Paper 00/13
October 2000


Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

# Hierarchical Document Clustering Using Automatically Extracted Keyphrases

**Steve Jones** **Malika Mahoui**
Department of Computer Science, University of Waikato
Private Bag 3105, Hamilton, New Zealand
{stevej, mmahoui}@cs.waikato.ac.nz

## Abstract

In this paper we present a technique for automatically generating hierarchical clusters of documents. Our technique exploits document keyphrases as features of the document space to support clustering. In fact, we cluster keyphrases rather than documents themselves and then associate documents with keyphrase clusters. We discuss alternative measures of similarity between 'soft-clusters' which seed Ward's hierarchical clustering algorithm, and present the resulting cluster hierarchies that we have produced for a large collection of scientific technical reports. We analyse the effect of the alternative similarity measures and suggest improvements to our technique.

## Introduction

The organisation of electronic documents into related groups or clusters can support users in accessing increasingly large volumes of information more effectively. However, manual grouping is prohibitively time consuming and expensive for most information providers. Ideally, information providers would be able to automatically arrange documents into sensible topic based groups or hierarchies, and then present these arrangements to users via appropriate browsing interfaces.

Some systems show that this is possible. For example, the New Zealand Digital Library (NZDL) (Witten et al., 1999a; Witten et al., 1998) provides a number of document collections in which structured browsing interfaces are automatically generated. However, grouping of documents in this manner is dependent on the 'owner' of the information providing structural information. The challenge of deriving this information automatically falls into the problem domain of document clustering (Willet, 1988).

In this paper we present an approach to automated document clustering that exploits keyphrases of documents as the attributes that underpin the clustering process and the labelling of clusters in browsing interfaces. We investigate alternative techniques for assessing the similarity of initial soft-clusters. We recognise that keyphrases are rarely provided with documents, and therefore apply Kea (Frank et al., 1999; Witten et al., 1999b), a machine learning system for automatic keyphrase extraction to the sub-problem of identifying suitable keyphrases to support clustering. We have applied our technique to a large scale, practical document collection—the NEC ResearchIndex (Lawrence et al., 1999) collection of scientific research papers.

In the next section we discuss a variety of approaches to document clustering. Following this we describe the process that we have developed for document clustering using the keyphrases provided by Kea. We then discuss the clusters that we have produced for the ResearchIndex collection, comparing alternative inter-cluster similarity measures. Finally we present our conclusions and a summary of the paper.

## 1 Automatic document clustering

Approaches to document clustering can be characterised in a variety of ways, and overviews of a range of techniques are provided by Willet (1988), Zamir *et al* (1997; 1998) and Jain *et al* (1999). One characteristic is the time at which the clustering process takes place, either before (static clustering) or during (dynamic clustering) a user's access to the collection of documents concerned. Static clustering (Jardine and van Rijsbergen, 1971; Salton, 1971) has several advantages: given that the arrangement of clusters (how many, their sizes and so on) is known, user interface mechanisms can be tailored to suit that arrangement; no clustering occurs during user access and so system response times are not degraded; human experts can tailor automatically generated clusters; and clusters can be exploited to increase the performance of a retrieval engine with respect to user queries. However, as the size of document collections grows, so does the time required to cluster them, and when collections are evolving rapidly, constant re-clustering may be infeasible.

When clusters are intended to support access to what are essentially user-defined collections (such as query result sets), static clustering is not possible. In these cases, dynamic clustering organises a result set in the context of a submitted query. For such systems (Scatter/Gather (Cutting *et al.*, 1993; Hearst and Pedersen, 1996), for example), efficient performance is important to maintain response time and linear time algorithms have been investigated (Cutting *et al.*, 1993; Hearst and Pedersen, 1996; Zamir and Etzioni, 1998; Zamir *et al.*, 1997). Given the benefits of the two approaches, it is unsurprising that some systems combine them, such as that described by Silverstein and Pederson (1997).

A second characteristic of a clustering process is the structure that it produces. Hierarchical clustering algorithms (such as that of Ward (1963)), produce a tree which recursively divides the document space into related groupings. In these structures, documents, or sets of documents are placed at leaf nodes. At each step in the clustering process, child nodes within the hierarchy combine to make a larger cluster—it is the most similar clusters which are merged. With respect to user interaction, hierarchical clustering algorithms are good in

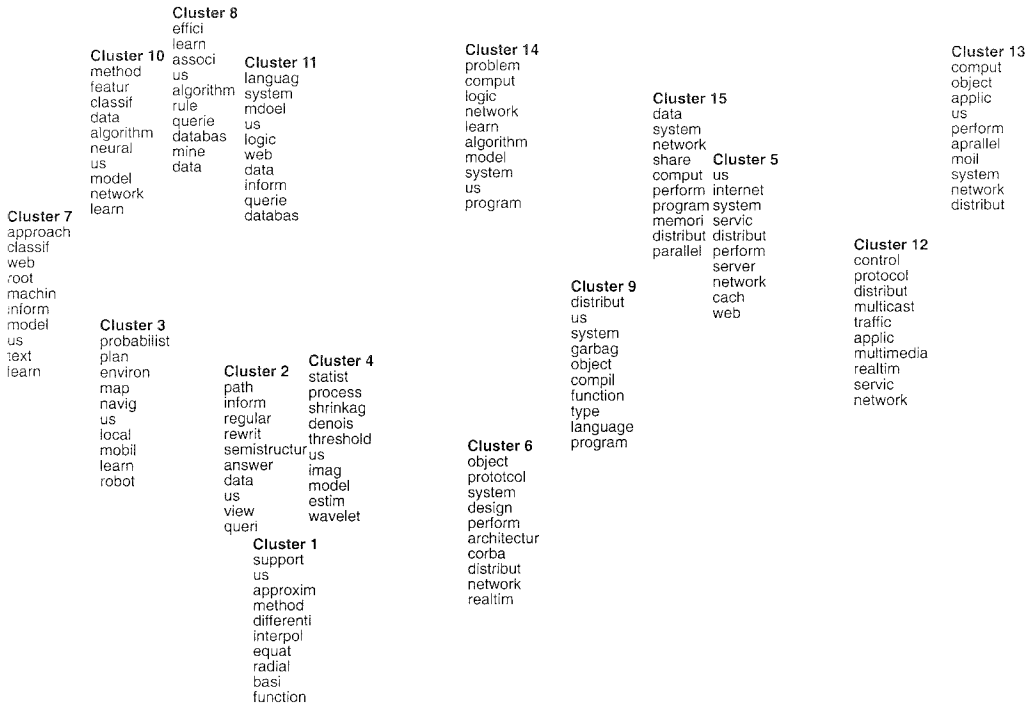


**Figure 1: Fifteen Clusters representing the ResearchIndex collection (Popescul *et al*, 2000)**

that users are often familiar with hierarchical information structures, and they can map directly to presentations of the structure of the information space. However, such hierarchies may be too deep or broad to sensibly facilitate interaction, although in this case, restricted representations that contain a fixed number of clusters from the top of the hierarchy can be produced. Flat partition algorithms are an alternative to the hiearchical approach. Here, initial clusters are selected (perhaps at random) and documents are assigned to those clusters according to some measure. Cluster centroids are recalculated and the process is repeated until some termination condition is met. The resulting clusters merely contain documents, and have no further structure, although the same process can be applied to each cluster in recursive manner. K-means (MacQueen, 1967) is the most basic example of this approach. A combination of both flat and hierarchical approaches can also be effective (Cutting *et al.*, 1992).

Common to most clustering methods is a procedure for computing the similarity between pairs of documents, and then using the similarity scores to arrange documents into sensible clusters. Commonly a vector space model (Salton, 1989) is used to represent the documents. This step involves the selection of a set of features (such as words) to represent the document and the application of a set of transformations to obtain new salient features, such as *tf.idf* scores. The vector representation of the documents is then used to define a similarity measure between them. The documents can then be clustered based on the similarity measure. Conventionally, documents have been represented by vectors of their constituent terms—a full text approach.

As collections grow, processing of the full text of documents becomes increasingly burdensome and less feasible. Clustering techniques face new challenges in proposing methods that scale well to manage the increasing size of web collections. One method is to reduce the dimensionality of the document feature space.

One dimensionality reduction method is to base the clustering on the citations extracted from documents, as in the ResearchIndex[1] system (Lawrence *et al.*, 1999; Popescul *et al.*, 2000). The most cited papers are selected to become the centroids of the "soft" clusters. Each document is then assigned to a cluster based on whether it is co-cited with the centroid of the cluster or not. On completion, the initial citations will represent the whole collection. A similarity measure used to compute the proximity between a pair of clusters is defined by the number of documents in common divided by the sum of the number of unique documents in the two clusters. Ward's hierarchical clustering algorithm is then used to produce a cluster dendogram, from which a cutoff point is derived giving a set of 15 clusters. To label the clusters, the titles of the documents of each cluster are stemmed and the highest frequency words are used to characterise the clusters. Figure 1 shows the resulting clusters and labels.

In this paper we apply a new method for clustering ResearchIndex documents in which the features that guide document clustering are presented directly to users as cluster labels. Popescul *et al* (2000) and other clustering approaches use documents themselves as cluster centroids, around which other documents are organised by some attribute (such as co-citation). We take a different approach in which we cluster frequently occurring keyphrases extracted from a document collection. We believe that keyphrases are good

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

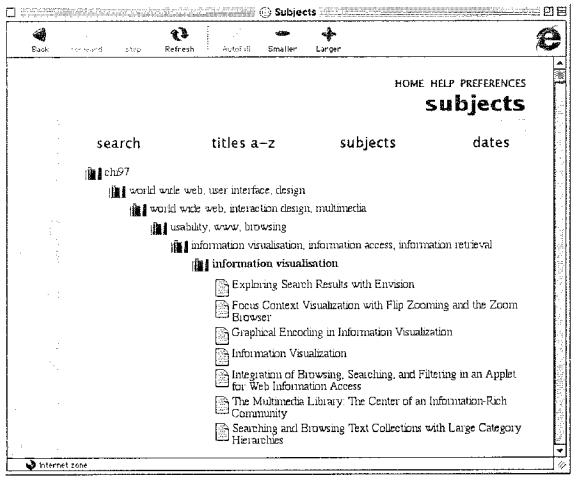


**Figure 2: An automatically generated browsable cluster hierarchy, exploiting keyphrase-based clusters**

descriptors of topics covered within, and therefore sensible subspaces of, the full document space. Once clustering is complete, the final stage of our method is to associate documents with keyphrase clusters. By using keyphrases as representative features by which the document space can be organised we substantially reduce the dimensionality of the document feature space to a manageable size, even for very large collections.

Our work is similar to Grouper (Zamir and Etzioni, 1998; Zamir and Etzioni, 1999) which uses a clustering technique based on identifying the phrases that are common to groups of documents. Grouper then presents lists of phrases to users for further investigation. Similarly, Maarek and Wecker (1994) used phrases as attributes of a document's vector representation. More specifically they used word pairs in order to help organise documents into what they called "dynamic bookshelves".

## 2 Phrase-based hierarchical clustering

Given a collection of source documents, the clustering process that we have developed generates hierarchical document clusters that support interactive browsing by end-users. An HTML-based example from the NZDL can be seen in Figure 2.

It is important to emphasise that our approach clusters *keyphrases* with which documents can later be associated, rather than the more common approach of clustering *documents* which are later labelled with keyphrases or words. The process can be controlled through several parameters, such as the total number of clusters required in the completed hierarchy, a cutoff level up to which clusters should be created, and the particular clustering algorithm to apply.

### 2.1 Approach

To begin we use the Kea system (Frank *et al.*, 1999; Witten *et al.*, 1999b) to extract keyphrases from each document in the collection to be clustered. Kea uses machine learning techniques to build a model of keyphrases in a collection, and applies that model to identify likely keyphrases within document text. Keyphrase stems output by Kea for each document are combined with author-specified keyphrases (where available) into a single list ranked in descending order of the number of documents to which they are allocated. Indexes which relate phrase stems and documents are also created.

The top $n$ keyphrases are selected from the list to determine how many leaf nodes will appear in the cluster hierarchy. These form the centroids of 'soft clusters' between which similarity scores are computed. We currently compute similarity scores in two ways. In the first, documents to which each of the $n$ keyphrases have been allocated are identified and treated as the attributes by which cluster similarity is determined. Similarity scores for each pair of resulting clusters are computed using the following equation, where $C_{p1}$ and $C_{p2}$ represent two clusters of which p1 and p2 are the respective centroids, and $D_{p1}$ and $D_{p2}$ are the sets of documents to which keyphrases p1 and p2 have been allocated respectively:

$$Similarity(C_{p1}, C_{p2}) = \frac{|D_{p1} \cap D_{p2}|}{|D_{p1} \cup D_{p2}|} \qquad (1)$$

This is the number of documents in common divided by the sum of the number of unique documents in the two clusters. If the two keyphrases had exact co-occurrence the ratio would be 1, if they had no documents in common the ratio would be 0. This measure follows that used by Popescul *et al* (2000) and Zamir and Etzioni (1998). It reflects the view that clusters are document-centric rather than topic-centric—that is, they organise similar documents, rather than similar topics.

An alternative view is that clusters should organise topics with which documents are then associated. We have applied this second view by using keyphrases which have also been allocated to the same documents as the soft-cluster centroids. These sets of co-allocated keyphrases are then the attributes by which similarity is determined. Consequently, we compute inter-cluster similarity based on the number of phrases that they have in common using the following equation. As before $C_{p1}$ and $C_{p2}$ represent two clusters of which p1 and p2 are the respective centroids. $P_{p1}$ and $P_{p2}$ are the sets of keyphrases co-allocated with centroids p1 and p2.

$$Similarity(C_{p1}, C_{p2}) = \frac{|P_{p1} \cap P_{p2}|}{|P_{p1} \cup P_{p2}|} \qquad (2)$$

Using the two equations we prepare dissimilarity matrices where each entry is subtracted from 1, to use as input to Ward's hierarchical clustering algorithm (Ward, 1963) as implemented in the multivariate analysis package of the R statistical package[2]. The output from this process is a concise representation of the resulting hierarchical structure.

This representation is then used to create displays and interactive versions of the cluster hierarchy, in which

[2] http://www.stat.auckland.ac.nz/rproj.html

centroid keyphrases are augmented by the documents to which they have been allocated. Documents may appear in more than one cluster and each cluster will contain at least one document.

In their basic form these hierarchies contain data only at their leaf nodes. However to get to the leaf nodes (documents) end-users must navigate down the hierarchy. Therefore, our system generates labels at each branching point of the hierarchy which describe the content of each of the branches or subtrees. These labels take the form of a subset of keyphrases found within a subtree. Our system uses the three most common keyphrases to be found in a subtree (as shown in Figure 2).

### 2.1.1 Test Collections

Figure 2 shows an extract from the product of our clustering process on a small test collection of 280 documents from the Proceedings of ACM CHI 97 (1997), using 50 soft clusters. This collection is useful for reasoning about and illustrating our approach, but we are concerned with clustering large, practical Web collections. To this end we have applied our technique to 291,397 documents indexed by the ResearchIndex system (Lawrence *et al.*, 1999). This collection has a broad coverage of scientific literature with an emphasis on Computer Science research papers. Some existing work on clustering this collection, against which we can compare our efforts, has previously been carried out (Popescul *et al.*, 2000).

For the ResearchIndex collection we used a pre-existing Kea model, derived from a large collection of Computer Science Technical Reports. From each document we took the stems of the five highest scoring Kea keyphrases where possible—some documents might have fewer than five. No author keyphrases were available, resulting in a total of 48990 keyphrase stems being extracted for the collection.

The stems were ordered according to the number of documents they were extracted from, and the 1000 most frequently extracted stems were selected. These formed the centroids of 1000 soft clusters. Two dissimilarity matrices were then computed, using each of the equations shown above. Ward's algorithm was then applied to each matrix to produce cluster hierarchies, for which each subtree was then labelled.

### 2.1.2 Results

Our first observation is that there is little correlation between the absolute cluster-pair similarity values produced by the two similarity measures presented. This can be seen in Figure 3, which shows the scores from Equations 1 and 2 plotted against each other, with pairwise repetitions removed. Both equations produce, on average, very low scores. With pairwise repetitions removed, the mean for Equation 1 is 0.0004 (sd = 0.0034) and for Equation 2 the mean is 0.0578 (sd = 0.0743). A larger number of distinct similarity values are produced by Equation 2 (35488) than Equation 1 (10139), although these are only 7% and 2% of the total number of similarity values, respectively.

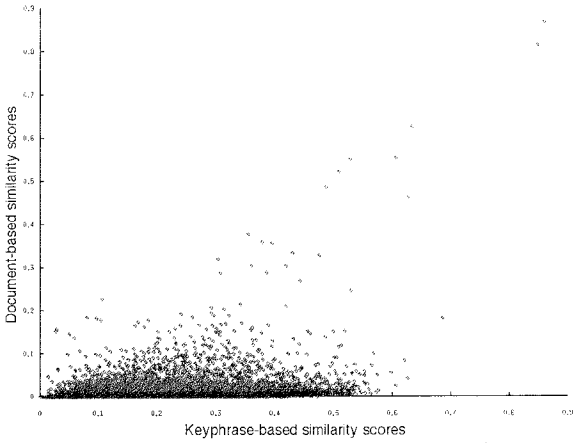


**Figure 3: Plot of keyphrase vs document based similarity scores for all cluster pairs**

However, the clustering algorithm exploits relative pairwise similarities rather than absolute values (that is, the most similar pair of clusters is merged). Although it might be the case that the similarity measures differ with respect to the values produced, they may produce similar ranking of cluster-pair similarity. Spearman's Rho (rank order correlation coefficient) was calculated for the cluster-pair similarity values generated by each equation. The value of rho (0.5596) indicates a positive correlation and is significant at the p=0.05 level. We should be cautious because the number of ties in the rankings, but this value is an indicator that the two equations produce rankings of inter-cluster similarities that are more similar than dissimilar.

Nevertheless, when we consider the cluster hierarchies generated, we see a substantial difference between the two structures, both in shape and in distribution of keyphrases. We have imposed a cutoff at 15 clusters for both hierarchies and these are shown in Figures 4 and 5. For each cluster we show three items: the step in the clustering process at which the cluster was created; up to the 20 most frequently occurring keyphrases from the underlying subtree of the hierarchy; and the number of documents associated with that cluster. The first ten phrases are highlighted to ease comparison against those of Popescul *et al* shown in Figure 1. Some keyphrases appear strange, such as "f7.75 f7.75 f7.75". Further investigation reveals them to be document formatting codes which conform to Kea's model of valid keyphrases.

When Equation 1 is applied (Figure 4) the majority of the clusters are highly specific, containing only one or two keyphrases. There is a tendency for each cluster subtree to contain one very specific branch, and one large branch. One cluster (985) contains the 20 most frequent keyphrases for the collection as a whole, and consequently covers more than 60% of all documents. Each of the other clusters covers less than 3% of all documents.

The result of applying Equation 2 (Figure 5) is a marked contrast. Branches of the hierarchy are more evenly weighted both in terms of the distribution of keyphrases and coverage of all documents. Almost all clusters contain 20 or more keyphrases. Again, one cluster (984) is heavily dominated by the most frequently allocated

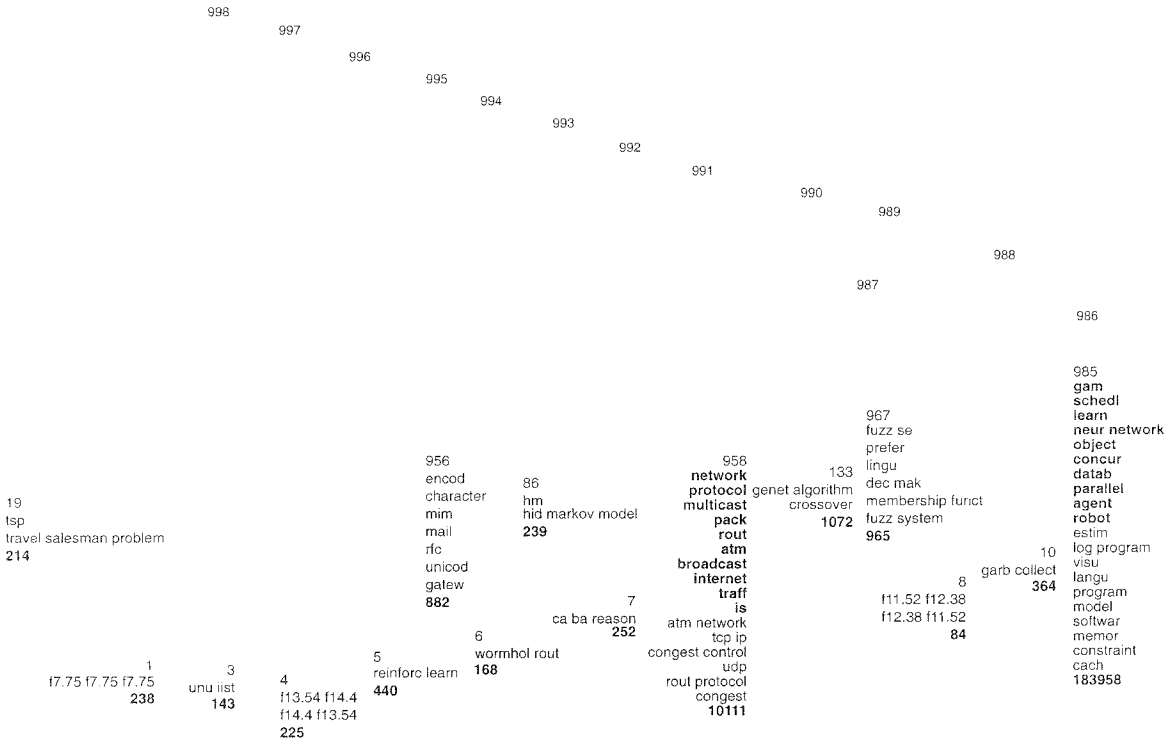


**Figure 4: top 15 clusters of the ResearchIndex hierarchy generated via document set-based similarity scores**

keyphrases and is the largest cluster in terms of the number of documents covered. Whereas many of the 15 clusters shown in Figure 4 are created very early in the clustering process, those in Figure 5 are the last clusters to be created.

## 3 Discussion

Whichever similarity equation is applied, most of the initial clusters are computed to be of little similarity (see Figure 3). When we consider the number of documents to which each keyphrase has been allocated, we find that the most commonly allocated keyphrase is the only one allocated to more than 1% of all documents. Therefore, the likelihood of an intersection between the sets of documents to which any two keyphrases have been allocated is very low. This produces consistently low similarity scores by Equation 1 (document-centric approach). However, this approach produces very specific initial soft-clusters which one might therefore expect to be quite distinct.

Equation 2 broadens the scope of a soft-cluster by considering co-allocated keyphrases. This notionally makes the clusters more general, and as we might expect it increases the likelihood of cluster overlap.

The analysis of the results produced through the clustering steps using each of the approaches reveals some interesting observations.

1. Keyphrases specific to a very small group of documents get clustered in the same way, relatively early in the clustering process for both approaches. For example we have the pair (f7.75 f7.75, f7.75) forming cluster 1 with both approaches; and the pair (travel salesman, travel salesman problem) forming cluster 2 with approach 1 and cluster 3 in approach 2.

2. Keyphrases associated with larger number of documents tend to be clustered much earlier by approach 2 than by approach 1. Indeed, when using approach 2, we found that in the first 100 clusters, 95 keyphrases were among the 100 most frequently allocated keyphrases. With approach 1, only 23 keyphrases that contributed in the first 100 clusters were among the first 100 most frequent keyphrases. For example the most frequent keyphrase stem "gam" is part of the cluster 8 with approach 2 while it is part of cluster 337 with approach 1.

3. Clusters already composed of the most frequent attributes tend to join more quickly with approach 2 than with approach 1. For example, using approach 2, cluster 8 (gam, network) is joined with keyphrase "model" ($18^{th}$ in the list of most frequent keyphrases). When considering approach 1, the first cluster merging the most frequent keyphrases is cluster 29 (learn, parallel); and cluster 29 is merged much later to contribute in cluster 175.

4. Keyphrases that are associated with relatively few documents tend to be clustered much earlier by approach 1 than by approach 2. With approach 1, 177 keyphrases, that contributed in the first 100 clusters, are among the list of keyphrases varying from the $100^{th}$ most frequent keyphrase to the 991st most frequent keyphrase. When using approach 2 this number falls down to 65 keyphrases; and most of them fall in the range [100,300] of the frequent keyphrases ordered list.

5. Clusters already composed of the least frequent attributes tend to merge later with approach 2 than with approach 1. Indeed, with approach 2, we found that 35

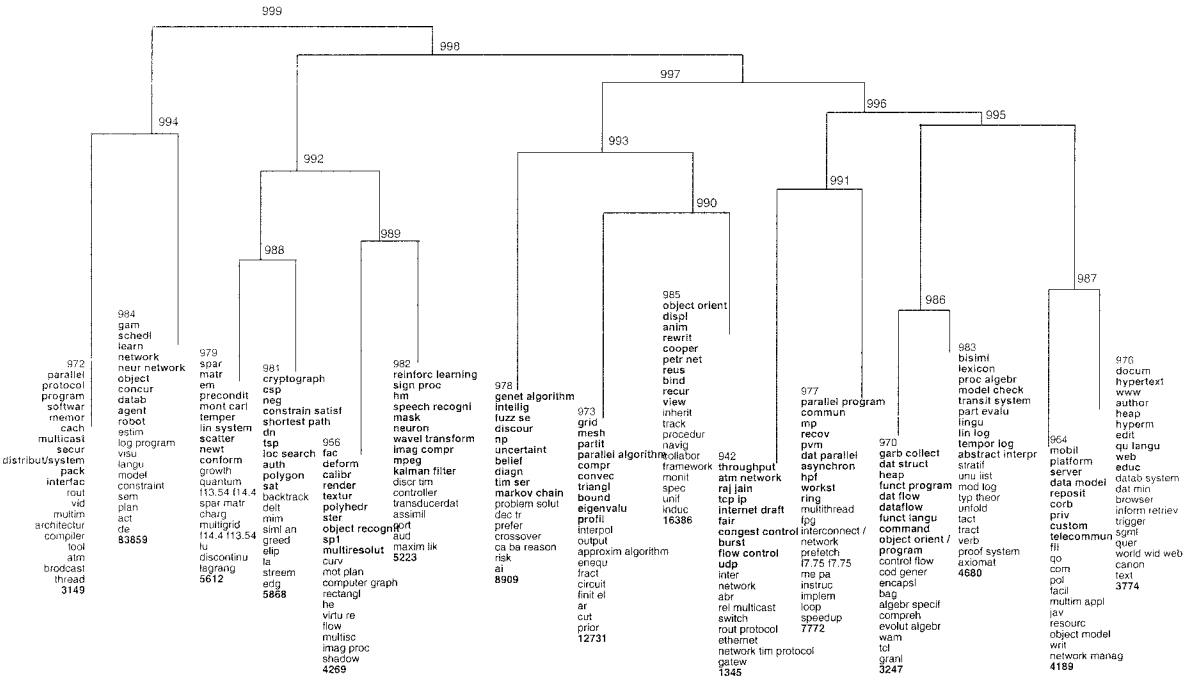


**Figure 5: top 15 clusters of the ResearchIndex hierarchy generated via keyphrase set-based similarity scores**

already composed clusters contributed to form new clusters in the first 100 steps. Only three of them included keyphrases that are at least 100th in the frequent keyphrases ordered list. With approach 1, the number of already formed clusters that joined in the first 100 steps are 3 and all include keyphrases that are at least 100th in the frequent keyphrases ordered list.

From observation 1, we conclude that very specific groups of documents are detected as such by both approaches. As a consequence of observations 2 and 3, these specific clusters will be absorbed by generic clusters formed early with approach 2. On the other hand, as a consequence of observations 1, 4 and 5, specific clusters are well distinguished from other clusters at early stage of the clustering process.

In fact Ward's minimum variance method is known to be in favour of small clusters and is very sensitive to outliers (Milligan, 1980). Approach 2 seems to diminish these drawbacks but at the same time it creates clusters like cluster 984 which might be too generic to be useful. A deeper cut off will not resolve the problem because these generic clusters have been created early in the clustering process and kept growing. This is not the case for approach 1 where the most frequent keyphrases are located in cluster 985 which is the next cluster to be broken down for a deeper cut off, and from observation 5 we know they are likely to be part of different clusters in fewer back track steps.

Approach 1 seems to emphasise Ward's method pitfalls. One way to overcome this drawback is to discard the very specific small clusters, by putting some threshold value on the number of keyphrases that each cluster should have. We have applied this filter by imposing a minimum of 3 phrases for a cluster to be considered as one of the 15 clusters and we obtained the results of Figure 6. The branches of the clusters that meet the threshold value are much more evenly weighted.

Cluster labels can show the similarity, or differentiate between clusters, which aids users in accessing them effectively. We believe that our keyphrase labels are more effective than the title word labels used by Popescul *et al* in discriminating between clusters. This is because each keyphrase appears only once in the hierarchy, and so never appears in more than one hierarchy label, whereas the generic word 'perform', for example, appears in four of the 15 clusters of Figure 1. Because of this, our approach is less good at communicating overlaps between clusters, although we believe that users will be able to establish them through interpretation of cluster labels.

Popescul *et al* (2000) reduce the dimensionality of a 150,000 item document space to a 474 by 475 symmetric similarity matrix, by excluding a large number of documents, creating soft-clusters and calculating similarities. We reduce a 291,000 item space to a 1000 by 1000 symmetric matrix, although we need to further investigate whether we need to use so many initial soft-clusters, given the infrequency with which many of their keyphrase centroids are allocated to documents.

Our approach suffers from some limitations. By picking the top *n* keyphrases to act as surrogates for initial document clusters it is possible that there will be some documents that are inaccessible through the resulting hierarchy. For the CHI97 collection 80 of the 280 documents were not allocated to any of the 50 clusters produced. This occurs in cases where none of a document's keyphrases occur in the top *n* positions in the list. In fact, it is possible that some documents have no keyphrases at all—none that are user specified and none that Kea was able to extract. Again these documents will be excluded from the hierarchy. Our solution is to

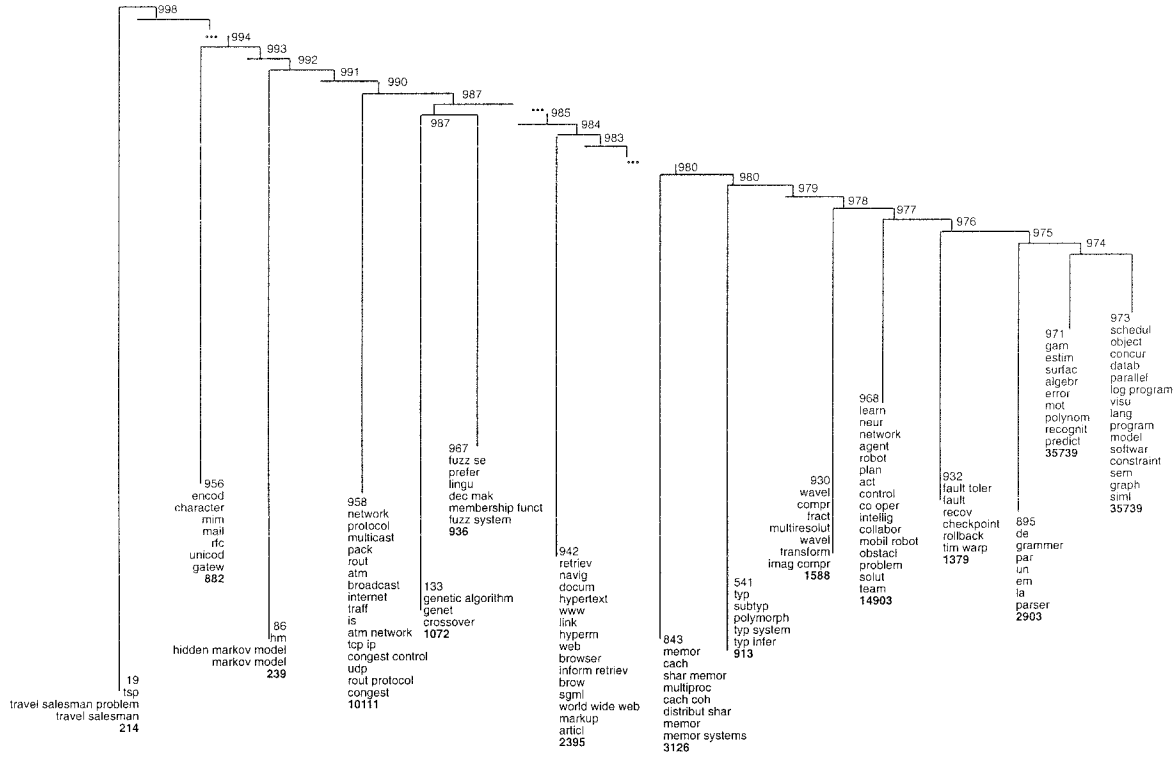


**Figure 6: amended hierarchy for approach 1 where each selected cluster has at least 3 keyphrases**

introduce a 'Miscellaneous' cluster into which such documents are placed.

Keyphrases that are allocated to many documents are less good at discriminating between documents than those that are associated with few. We have observed that more frequently occurring keyphrases are more generic. The result of this is that users will likely be faced with cluster labels that aren't specific enough to help them to decide which clusters are of interest. However, they will get good coverage of the documents in the collection. Conversely, if we pick less frequent keyphrases users might be faced with too specific cluster labels, and poor coverage of the documents. One approach which might resolve both of these issues is to continue to select phrases from the list until all possible documents have been covered. Another might be to select a mixture of general and specific keyphrases.

The effect and interaction of a range of variables such as the overall number of clusters required, the coverage of the document collection that is acceptable, and the appropriate generality of the cluster labels require further investigation. In particular this investigation will need to focus on users requirements and preferences when interacting with such automatically generated clusters.

## 4 Future work

We are encouraged by the nature of the clusters produced by our approach. Subjectively, they appear to be at least as good representations of the ResearchIndex topic coverage as those produced by a citation based approach, and merit further investigation. Clearly, the next step will be to evaluate the clusters. There are a variety of questions to be asked: do potential users perceive the items in cluster labels to be strongly related, do they provide a good indication as the the documents in the clusters, and how well related are the documents in a given cluster. These will be answered by a combination of subjective and objective approaches.

Additionally we will be looking at the effect of further inter-cluster similarity measures. Specifically we will investigate how a standard similarity measure (such as the cosine measure) can be applied to initial soft clusters. There is evidence that a keyphrase-vector based cosine measure can be effective (Jones and Staveley, 1999) in other situations.

## Conclusions

We have presented a technique which uses keyphrases of documents as the attributes of a document space upon which clustering is based. The keyphrases are automatically extracted from documents. We have used Ward's hierarchical clustering algorithm, and proposed two methods for seeding the algorithm. The first exploits similarities between the sets of documents to which keyphrases have been allocated. The second exploits similarities between sets of co-occurring keyphrases.

Even though the cluster-pair similarity values are well correlated for each method, we have found that they produce hierarchies of very different natures. This is with respect to shape, distribution of keyphrases and distribution of documents. In situations where highly specific clusters are required towards the top of the hierarchy the first method should be used. The second method (keyphrase co-occurrence) should be used when more generic clusters and even document distributions are required at higher levels of the cluster hierarchy.

Whichever method is used, the most frequently allocated keyphrases will tend to produce one or two large clusters.

We have also suggested that sets of keyphrases are suitable candidates for cluster labels, and proposed that within any given branch of a cluster hierarchy the most frequently allocated keyphrases are good label candidates. This approach is particularly useful for discriminating between clusters, but less effective in communicating similarity between clusters. Finally, our approach substantially reduces the dimensionality of the document space for clustering purposes, increasing the feasibility of clustering very large document collections.

## Acknowledgements

Thank you to Gordon Paynter for advice about the Kea system. The browsable hierarchy tools for the NZDL have been developed by Stefan Boddie and David Bainbridge. Many thanks to the ResearchIndex team for making their document collection available for analysis.

## References

1997 *Proceedings of CHI'97: Human Factors in Computing Systems*, ACM Press.

Cutting, D. R., D. Karger and J. Pedersen 1993 *Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections*. Proc. of SIGIR'93: the 16th International Conference on Research and Development in Information Retrieval, pp. 126-135.

Cutting, D. R., D. Karger, J. Pedersen and J. W. Tukey 1992 *A Cluster-Based Approach to Browsing Large Document Collections*. Proc. of SIGIR'92: the 15th International Conference on Research and Development in Information Retrieval, ACM Press, pp. 318-329.

Frank, E., G. Paynter, I. Witten, C. Gutwin and C. Nevill-Manning 1999 *Domain-specific Keyphrase Extraction*. Proc. of the Sixteenth International Joint Conference on Aritificial Intelligence, Morgan-Kaufmann, pp 668-673.

Hearst, M. and J. Pedersen 1996 *Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results*. Proc. of SIGIR'96: the 19th International Conference on Research and Development in Information Retrieval, pp. 76-84.

Jain, A. K., M. N. Murty and P. J. Flynn 1999 *Data Clustering: a Review*. ACM Computing Surveys 31/3, pp. 264-323.

Jardine, N. and C. J. van Rijsbergen 1971 *The Use of Hierarchical Clustering in Information Retrieval*. Information Storage and Retrieval 7, pp. 217-240.

Jones, S. and M. Staveley 1999 *Phrasier: a System for Interactive Document Retrieval Using Keyphrases*. Proc. of SIGIR'99: the 22nd International Conference on Research and Development in Information Retrieval. M. Hearst, F. Gey and R. Tong. Berkeley, CA, ACM Press, pp. 160-167.

Lawrence, S., K. Bollacker and C. L. Giles 1999 *Indexing and Retrieval of Scientific Literature*. Proc. of 8th International Conference on Information and Knowledge Management, CIKM 99, pp. 139-146.

Maarek, Y. and A. J. Wecker 1994 *Automatically Organizing On-line Books Into Dynamic Bookshelves*. Proceedings of the 1994 RIAO Conference (RIAO'94),

MacQueen, J. 1967 *Some Methods for Classification and Analysis of Multi-Variate Observations*. Proc. of the Fifth Berkeley Symposium on Math, Statistics and Probability. L. M. LeCam and J. Neyman, Berkeley University of California Press, pp. 281-297.

Milligan, G. W. 1980 *An Examination of The Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms*. Psychometrika 45, pp. 325--342.

Popescul, A., G. W. Flake, S. Lawrence, L. H. Ungar and L. C. Giles 2000 *Clustering and Identifying Temporal Trends in Document Databases*. Proc. of the IEEE Advances in Digital Libraries 2000 (ADL2000), pp 173-182.

Salton, G. 1971 *Cluster Search Strategies and the Optimization of Retrieval Effectiveness*. The SMART Retrieval System. G. Salton, Prentice-Hall, pp. 223-242.

Salton, G. 1989 *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley.

Silverstein, C. and J. Pedersen 1997 *Almost Constant Time Clustering of Arbitrary Corpus Subsets*. Proc. of SIGIR'97: the 20th International Conference on Research and Development in Information Retrieval, pp. 60-67.

Ward, J. H. 1963 *Hierarchical Grouping to Optimize Objective Function*. Journal of ASA 58, pp. 236-244.

Willet, P. 1988 *Recent Trends in Hierarchical Document Clustering: a Critical Review*. Imformation Processing and Management 24, pp. 577-597.

Witten, I. H., R. J. McNab, S. Jones, M. Apperley, D. Bainbridge and S. J. Cunningham 1999a *Managing Complexity in a Distributed Digital Library*. IEEE Computer 32/2, pp. 74-9.

Witten, I. H., C. Nevill-Manning, R. McNab and S. J. Cunningham 1998 *A Public Library Based On Full-Text Retrieval*. Communications of the ACM. 41/4, pp. 71-75.

Witten, I. H., G. W. Paynter, E. Frank, C. Gutwin and C. G. Nevill-Manning 1999b *KEA: Practical Automatic Keyphrase Extraction*. Proc. of Digital Libraries '99: The Fourth ACM Conference on Digital Libraries. E. A. Fox and N. Rowe. Berkeley, CA, ACM Press, pp. 254-255.

Zamir, O. and O. Etzioni 1998 *Web Document Clustering: a Feasibility Demonstration*. Proc. of SIGIR'98: the 21st International Conference on Research and Development in Information Retrieval, ACM Press, pp. 46-54.

Zamir, O. and O. Etzioni 1999 *Grouper: A Dynamic Clustering Interface to Web Search Results*. Computer Networks and ISDN Systems 31/11-16, pp. 1361-1374.

Zamir, O., O. Etzioni, O. Madani and R. M. Karp 1997 *Fast and Intuitive Clustering of Web Documents*. Proc. of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97), pp. 287-290.