ELSEVIER

Decision Support Systems 27 (1999) 67–79

Decision Support Systems

www.elsevier.com/locate/dsw

Document clustering for electronic meetings: an experimental comparison of two techniques

Dmitri G. Roussinov *, Hsinchun Chen [1]

Department of MIS, Karl Eller Graduate School of Management, University of Arizona, McClelland Hall 430ww, Tucson, AZ 85721, USA

Abstract

In this article, we report our implementation and comparison of two text clustering techniques. One is based on Ward's clustering and the other on Kohonen's Self-organizing Maps. We have evaluated how closely clusters produced by a computer resemble those created by human experts. We have also measured the time that it takes for an expert to ''clean up'' the automatically produced clusters. The technique based on Ward's clustering was found to be more precise. Both techniques have worked equally well in detecting associations between text documents. We used text messages obtained from group brainstorming meetings.

Keywords: Group decision support systems; Text document clustering; Empirical study; Self-organizing maps; Neural networks; Cluster analysis

1. Introduction

The rapid proliferation of information available in electronic format has turned a dream of creating an information-rich society into a nightmare of information overload. For example, users currently foraging for information on the World Wide Web receive an average of more than 30 000 documents in answer to a query. Many researchers believe that turning information abundance into a useful digital library requires developing new technologies.

Summarization and visualization tools can help users understand the information that is contained in a large collection of documents. Modern visualization techniques available for digital libraries comprise two major components: the *agglomeration* component that identifies clusters of similar documents and the *summarization* component that presents an automatically computed overview of the documents in each cluster. By applying them, a user does not need to wade through all the documents in a collection one-by-one but may grasp a high-level picture instead. Clustering and classification are believed to be important techniques for semantic analysis in a new generation of digital libraries [2].

Clustering historically has been perceived by researchers in various domains to be a tool of discovery. It partitions a set of objects into non-overlapping subsets called clusters such that the objects inside each cluster are similar to each other and the objects from different clusters are not similar. The set of non-overlapping clusters is called a *partition*.

In this study, we have compared two documents clustering techniques using data obtained from electronic meeting sessions described in more detail in Ref. [18]. Efficient organizing of electronic meeting

* Corresponding author. Present address. School of Information Studies, Syracuse University, 4-234 Center for Science and Technology, Syracuse, NY 13244-4100, USA. Tel.: +1-315-443-1892; fax: +1-315-443-5806; E-mail: droussin@syr.edu

[1] E-mail: hchen@bpa.arizona.edu.

0167-9236/99/$ - see front matter
PII: S0167-9236(99)00037-8

comments is itself a rewarding task. Electronic meeting support has been proven to have great impact on productivity of group discussions [17]. Because participants in electronic meetings can generate hundreds of comments in an hour, the task of categorizing and organizing them is very time consuming. Intelligent agents may significantly reduce the cognitive load of meeting participants by automatically organizing documents into clusters, even if manual post-processing still is deemed necessary.

We have evaluated two clustering algorithms: Ward's clustering [26] and Kohonen's Self-organizing Maps [13]. Ward's clustering falls into the category of statistical clustering techniques. A self-organizing map is an unsupervised two-layer neural network. Although, both techniques may serve to cluster data, they do it in different ways. Statistical techniques proceed by pair-wise comparison of objects. Neural networks proceed by a process called *learning*. Neural networks are believed to possess some particularly valuable properties, since they are patterned after associative neural properties of the brain. The rationale for our research was that it would be useful to compare the techniques of these two very different approaches and see which performed better in the domain of unstructured text. It is of interest that the NSF/ARPA/NASA-funded Illinois Digital Library Initiative project [24] has adopted SOM for textual document categorization and visualization.

We supply more details on both techniques in Section 2. Then, in Section 3, we describe our implementations of the techniques. The experiment design follows in Section 4. Section 5 presents our findings. Section 6 describes conclusions and future research.

2. Literature review

2.1. Clustering text documents

Everitt [7] defined a cluster as ''a set of entities which are alike, and entities from different clusters are not alike.'' An example of an early study on clustering in Information Science is the work by Jardine and van Rijsbergen [12]. The idea behind clustering was that if certain documents match a user query, the documents in the same cluster also are likely to be relevant.

A good overview of the use of clustering applications in information retrieval has been done by Rasmussen [19]. She identified the following major problems with applying various clustering techniques in the text analysis domain:

1. Difficulty of assessing the validity of results obtained.
2. Selecting appropriate attributes for clustering.
3. Selecting an appropriate clustering method.
4. High cost in terms of computational resources.

This paper addresses each of those issues to a certain degree.

Recently, information visualization techniques have revived interest in clustering. The idea behind many of these techniques that are able to visualize large collections of documents is to agglomerate similar documents into clusters and present a high-level summary of each cluster. This way, the user does not need to go through similar documents or through entire documents in order to become familiar with the collection. This greatly reduces redundancy and cognitive demand. Examples of such visualization systems are Scatter/Gather [4], WebBook [1], and SenseMaker [25]. Hearst [9] gives a comprehensive overview of such systems and the ideas behind them.

2.2. Text categorization

The text clustering task resembles the text categorization task, which also has been extensively studied by information scientists. By definition, *Text Categorization* is the assignment of natural language texts to one or more predefined categories, based on their content. Examples of recent works in automatic text categorization are Refs. [5,28]. It has been shown that automated text categorization can be performed with 90% or greater accuracy in the cases of ''clean'' collections, of which Reuters [5] is an example.

However, the clustering task is more challenging, since there is no pre-existing set of categories created by human experts. The implication for machine learning is that, in contrast to the supervised techniques used in categorization, only unsupervised

techniques can be used for clustering. The clustering task has more degrees of freedom: not only assigning decisions but also decisions about how many clusters to create and what kinds of documents to assign to each cluster.

2.3. Evaluating clusters

Traditionally in Information Science, clustering techniques have been evaluated in conjunction with retrieval tasks. For example, Cutting et al. [4] and Hearst and Pedersen [8] evaluated the accuracy of clustering based on the proportion of relevant documents found in the largest cluster. We have found surprisingly few studies involving methodological evaluation of clustering techniques based on resemblance between the resulting partitions and clusters produced by human experts.

Sahami et al. [22] based their measurement on whether or not a pair of objects was put into the same class by human experts and by the system. For human expert judgments they used manually created categories existing in the Reuters collection. Zamir et al. [29] used a similar measurement to test clustering applied to a collection created by merging several smaller collections of Web documents on different topics. We have chosen to use similar metrics in our study. We provide more details in Section 4.

In prior studies on document clustering, the benchmark collections were created by merging documents on different topics. We are not aware of any other study involving evaluating document clustering techniques through experiments with human subjects. This makes our evaluation approach innovative.

2.4. Ward's clustering

Hierarchical agglomerating clustering (HAC) algorithms are the most commonly used method of document clustering [27]. These algorithms start with each document in a cluster of its own, iterate by merging the two most similar clusters, and terminate when some halting criterion is achieved.

One of the most popular HAC algorithms is Ward's clustering, proposed by statistician Ward [26]. Over time, it has been extensively used in various domains: astrophysics, pattern recognition, applied statistics, etc. In 1984, Murtagh proposed the reciprocal nearest neighbor approach (RNN) which is significantly faster than the straightforward implementation but produces identical results. The advantage is a resulting time complexity of $O(N^2)$ in comparison with $O(N^3)$ for the classical implementation. N represents the number of inputs, which in our case are text documents. Ward's clustering has been repeatedly applied for text analysis; El-Hamdouchi and Willett [6] is an example.

In our study, we also used the RNN approach, described in pseudocode in Fig. 1. By definition, two clusters, C1 and C2, are called RNNs if C1 is the nearest neighbor for C2, and C2 is the nearest neighbor for C1. The algorithm uses the inverse Euclidean distance between the centroids of the clusters as the measure of similarity between those clusters. The centroid of a cluster is computed by averaging the coordinates of all documents in the cluster. More details are found in Ref. [16], which also contains a proof that this algorithm terminates and produces output that does not depend on the order of selecting a cluster as the current cluster.

```
Form a cluster from each document.
Until only one cluster remains do
 Pick arbitrary cluster as Current Cluster;
 Found = False;
 Until not Found do
  find the closest neighbor to Current Cluster;
  if they are reciprocal nearest neighbors then
     Merge them;
     Found = True;
  otherwise
     Change Current Cluster to its nearest neighbor;
 end-do
end-do
```

Fig. 1. The pseudocode for Ward's clustering.

This algorithm runs until all documents are merged into a single cluster containing all of them and produces a balanced binary tree called a *dendrogram*. Each node in the dendrogram corresponds to a cluster obtained as a result of merging two other clusters corresponding to the child nodes. Since in our study we evaluated partitions, we converted the dendrogram into a partition. Murtagh [16] suggested a variance threshold technique for this purpose. The technique traverses the dendrogram tree and splits clusters associated with a node into two clusters associated with the corresponding child nodes. It stops when the average similarity between documents in each cluster and its centroid exceeds the specified threshold. Section 3 describes our modifications to this technique.

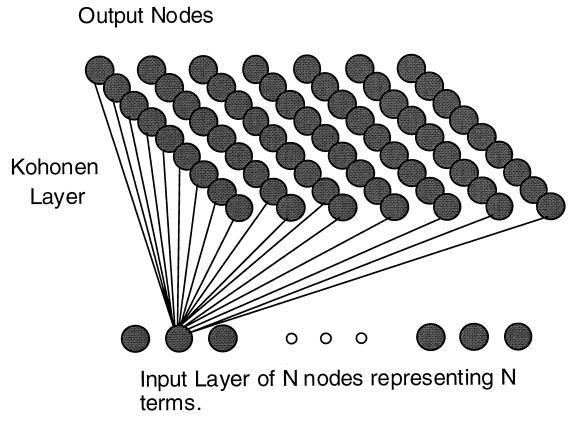


Fig. 2. Kohonen SOM topology.

2.5. Self-organizing maps

The Self-organizing Map, developed by Kohonen [14], is an unsupervised two-layer neural network used for clustering and dimension reduction. An advantage of SOM over other clustering algorithms is its ability to visualize high dimensional data using a two-dimensional grid while preserving similarity between data points as much as possible. It is a similar technique to Multidimensional Scaling (MDS) [11]. In SOM, each input node corresponds to a dimension. Each output node corresponds to a node in a two-dimensional grid. The network is fully connected in that every mapping node is connected to every input node with some connection weight. During the training phase, the inputs are presented several times in order to train the connection weights in such a way that distribution of output nodes represents distribution of input points. The network trains fully automatically, without any human intervention. The topology of the Kohonen SOM network is shown in Fig. 2. More details about the algorithm are given in Section 3.

Several recent studies adopted the SOM approach to textual analysis and classification [15]. Ritter and Kohonen [20] applied the Kohonen SOM to textual analysis in an attempt to detect the logical similarity between words from the statistics of their contexts. More recently, the Kohonen group has created and maintained a WEBSOM server that demonstrates its ability to categorize several thousand Internet newsgroup items [10]. The SOM-generated categories were found to be comparable to those generated by human subjects [18]. Chen et al. [3] applied multilayered SOM to categorize (classify) 110,000 Internet homepages according to their content.

Speed has been a major concern with SOM, especially in the text analysis domain, where vector dimensions may be very large. The research described in Ref. [21] proposed a modification of the original SOM algorithm that produces the same output but scales almost linearly with the size of the task. The approach takes advantage of the sparseness of the representation. This approach allows using SOM in real-time Digital Library applications such as interactive Web search.

3. Algorithms and implementations

This section explains in general terms how our automatic text clustering systems were implemented. Our process proceeds by the following steps:

1. Automatic indexing.
2. Selecting most discriminating terms.
3. Applying clustering technique: SOM or Ward's.

The purpose of automatic indexing is to identify the content of each textual document automatically by sets of associated features [23]. Features are words and phrases. Automatic indexing first extracts a set of all words and possible phrases that enter the document. Then, it removes words from a ''stop-

word'' list to eliminate non-semantic bearing words such as ''the'', ''a'', ''on'', and ''in''. Our automatic indexing program also creates phrases from adjacent words. In this research, we used the automatic indexing described in Ref. [18].

For computational efficiency and accuracy of representation, we preserved only the top 100 most frequently included terms in the collection. This approach works best with small collections consisting of short text messages, since it provides the greatest overlap in representations. Table 1 shows a list of the 20 most frequently appearing terms in the collection that we used. The average number of keywords preserved in the representation of a document was 3.7.

We used the Information Science community's most popular representation of documents: vectors in vector space [23]. Each coordinate in the vector space corresponds to a term. A term can be a single word or a phrase. If a term does not enter the document, the corresponding coordinate is set to 0. If a term enters the document, we set the coordinate to 1. Prior research [18] has shown this scheme to be adequate for electronic meeting messages.

3.1. Ward's clustering

We have found the variance threshold described in the literature [16] to be inadequate for using Ward's clustering in our task. The variance threshold approach often produced one big cluster, containing half of all the documents not necessarily similar to each other, and many small clusters, with 1–3 documents in each.

Table 1
The 20 most frequently occurring terms in the collection

0 Meetings	10 Networks
1 Meeting	11 Support
2 Technology	12 Notes
3 Collaborative Systems	13 Hardware
4 Information	14 Facilitators
5 Collaborative	15 Technologies
6 Distributed	16 Language
7 Systems	17 Wireless
8 Linear Thread Meeting	18 Network
9 Environments	19 Bandwidth

We devised a so-called *shared keyword rule* to alleviate the above problem. It followed the same recursive traversal procedure as the variance threshold but split the clusters only if there was no common keyword entering all documents in the cluster.

In more detail, the algorithm starts from the root of the dendrogram that Ward's clustering produces. The algorithm checks for the presence of a term (word or phrase) that all the comments below the root possess. If not, the algorithm assigns all the documents to two clusters according to the dendrogram. It then recursively checks each of the two obtained clusters in the same way. This way, the dendrogram influences only decisions on how to split the clusters, but not when to stop. At the end, documents in each cluster have at least one term in common. We adopted this approach because a similar one was adopted in the SOM described in Ref. [18]: the regions in the SOM were formed by merging map nodes that had the same most representative terms. We empirically found out that this approach resulted in a greater number of clusters of meaningful size (three or more documents). Since the comments in each cluster had at least one term in common, they were also likely to discuss similar issues.

We have also found another modification crucial to making the approach computationally tractable and scalable. In the text analysis domain, the dimensions of the vector space are large. In this study the input vector size was 100. If we represent the vector size by N, the time complexity of the straightforward similarity computation is $O(N)$, because Euclidean distance requires iterations through all the coordinates. Our modification requires only non-zero coordinates in a vector representation of documents to be stored, for example as a linked list. When distance is computed, the iteration cycle is organized in such a way that it goes through only non-zero coordinates. This changes the $O(N)$ complexity into $O(M)$, where M is the average number of non-zero coordinates in the document representation, which can be hundreds of times smaller than N.

3.2. Self-organizing maps

A sketch of a revised SOM algorithm for textual classification is summarized below. More details can be found in Ref. [18].

3.2.1. Initialize input nodes, output nodes, and connection weights

Create a two-dimensional map (grid) of M output nodes (say a 20-by-10 map of 200 nodes). Initialize weights $\boldsymbol{w}_{ij}$ from $\boldsymbol{N}$ input nodes to M output nodes to small random values. This way, each input node corresponds to a coordinate axis in the document vector space. Each output node is associated with a vector of weights $\boldsymbol{w}_{ij}$ so it can also be considered as a point in the input vector space.

3.2.2. Present each document in order

Describe each document as an input vector of $\boldsymbol{N}$ coordinates. Set a coordinate to 1 if the document has the corresponding term and to 0 if there is no such term. Each document is presented to the system several times.

3.2.3. Compute distance to all nodes

Compute Euclidean distance d_j between the input vector at time t, $\boldsymbol{x}_i(t)$, and each vector of weights $\boldsymbol{w}_{ij}$ representing an output node:

$$d_j = \sum_{i=0}^{N-1} \left(\boldsymbol{x}_i(t) - \boldsymbol{w}_{ij}(t)\right)^2$$

3.2.4. Select winning node j and update weights to node j* and its neighbors*

Select winning node j^*, which produces minimum d_j. Update weights to node j^* and its neighbors to reduce the distances between them and the input vector $\boldsymbol{x}_i(t)$:

$$\boldsymbol{w}_{ij}(t+1) = \boldsymbol{w}_{ij}(t) + \eta(t)\left(\boldsymbol{x}_i(t) - \boldsymbol{w}_{ij}(t)\right)$$

After such update, the nodes in the neighborhood of j^* become more similar to the input vector $\boldsymbol{x}_i(t)$. Here, $\eta(t)$ is an error-adjusting coefficient ($0 < \eta(t)$

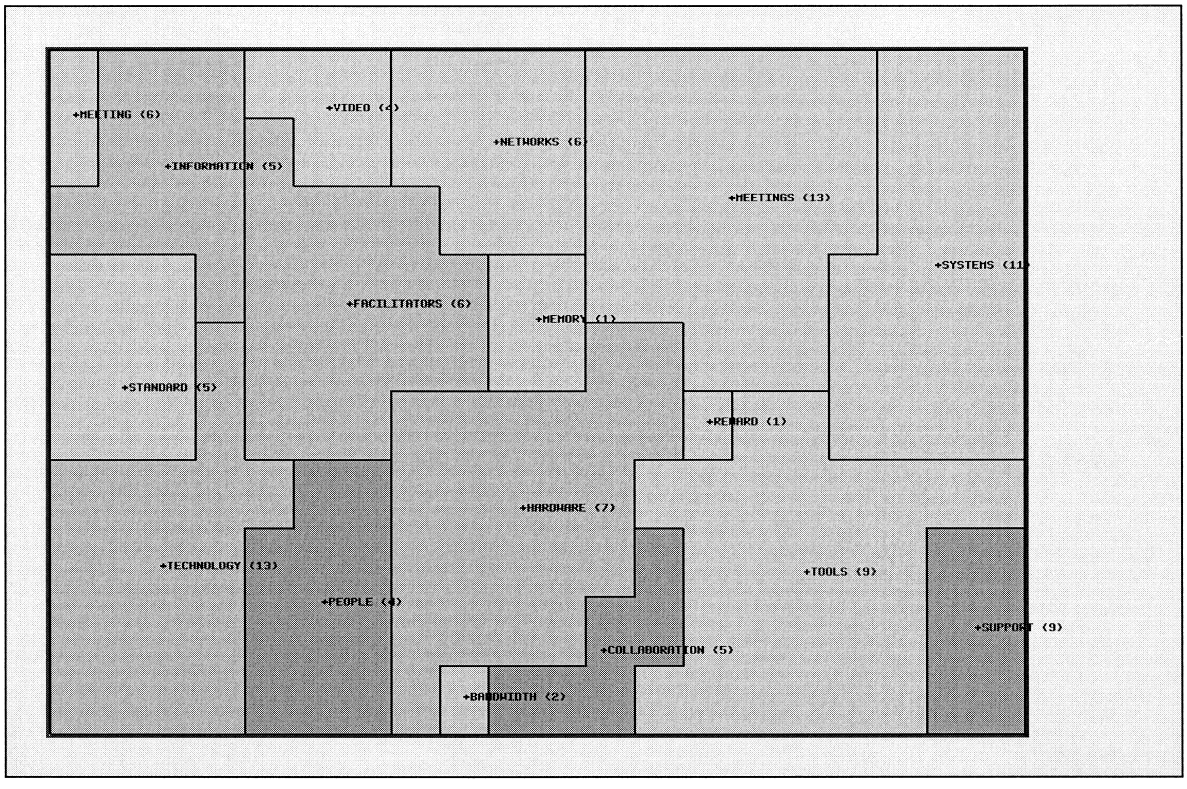


Fig. 3. The self-organizing map that we used for our study.

< 1) that decreases over time. Please see Ref. [13] for the algorithmic details of neighborhood selection and adjustment.

3.2.5. Label regions in map

After the network is trained through repeated presentations of all documents (each document is presented at least five times), assign a term to each output node by choosing the one corresponding to the largest weight (*winning term*). Neighboring nodes which contain the same winning terms are merged to form a concept/topic region (cluster). Assign each input document to the node with the closest vector of weights $\boldsymbol{w}_{ij}$. Consequently, the resulting map represents regions of important terms/concepts with the documents assigned to them. Concept regions that are similar (conceptually) appear in the same neighborhood. Similar documents are assigned into the same or similar concepts.

Fig. 3 shows the map that we generated and used in the current research. Each region represents a topic (or cluster). The number following each topic represents the number of documents assigned to each topic. Since in this study we evaluated only clustering properties, we converted the produced map into a partition before conducting experiments. The human subjects did not see the map itself nor the labels for the regions.

4. Evaluation experiment

4.1. Experiment design

We performed an experiment involving 17 human subjects. The subjects were volunteering business school students (10 graduate and 7 undergraduate) familiar with the topic of discussion and not involved in this research. Each subject received two text files that contained text comments grouped into topics (clusters).

Subjects were asked to re-arrange the comments in the file according to their own judgment. One file was the output from SOM; the other was the output from Ward's clustering. The order of file presentation was reversed for half of the subjects. It took 40–50 min on average for a subject to perform the task. Subjects worked on their own. We did not try to observe how subjects moved the comments around.

It should be noted that in this design subjects were given an initial set of clusters created by a computer, not text comments in random order. The final partitions created by the subjects therefore may have been influenced by our initial partitions. We deliberately designed our experiment in this way since we were interested in the amount of effort that it would take to change the partition suggested by a computer to one the user deemed adequate. This sequence simulates the semi-automatic classification of text collection that takes place in electronic meetings and other tasks where classification of text documents involves some additional manual work.

We used the output from an electronic brainstorming meeting containing 206 comments. The meeting participants discussed the issue ''The Future of GroupWare.'' Fig. 4 shows a portion of one of the files given to the subjects. Comments follow each other sequentially in a file from top to bottom, separated by at least one empty line. A special text string (''*** New Topic ***'') separates clusters from each other. Clusters did not have any labels

```
*** New Topic ***

Currently collaborative rooms are expensive to set up and maintain.
We need to be able to do portable environments in settings where we
would have normal meetings.  This would entail the use of wireless lan
technology that work beyond the line of site in the current
technology

Cellular technology hold great promise for having remote
Collaborative meetings but the reliability and band with need to be
Improved.

Repository technology will have to move toward an acceptable
Standard that will promote optimal sharing between various types of
Environments.  Don't make our own standards.

*** New Topic ***

Understanding how our technology encapsulates specific cultural
Expectations -- and learning which technologies are appropriate in
Which cultures.
```

Fig. 4. Sample EBS comments.

associated with them; since the outputs from both clustering algorithms looked structurally the same subjects were unable to distinguish between the techniques involved.

We treated each textual comment entered by a meeting participant as an independent document. EBS comments exhibit some unique characteristics and often contain typos, abbreviations, and incomplete sentences. Typically, a 1-h session with a dozen or so participants generates several hundred comments. Since manual clustering of the entire session would be very time consuming, we asked a human expert to choose 80 comments falling into approximately 8–10 categories (topics) and used only the selected comments in our experiment.

4.2. Metrics used

As we describe in Section 2, to measure the quality of clusters obtained automatically, we used their ''closeness'' to clusters created by humans in terms of the number of wrong and missed associations. The definitions below help to explain this measure.

We call a partition created by an expert a *manual partition.* An *automatic partition* is one created by a computer. Inside any partition, an *association* is a pair of documents belonging to the same cluster. *Incorrect associations* are those that exist in an automatic partition but do not exist in a manual partition. *Missed associations* are those that exist in the manual partition but do not exist in an automatic partition. We define *clustering error* as:

$$\mathrm{CE} = \frac{E}{P_t}$$

where P_t is the total number of possible pairs of documents: $P_t = 1/2\,D(D-1)$. E represents the total number of incorrect and missed associations: $E = E_i + E_m$.

This measure favors small partitions. To provide less dependence on the size of both partitions, we also used a *normalized clustering error*, expressed as:

$$\mathrm{NCE} = \frac{E}{A_t}$$

Here, A_t is the total number of all associations in both partitions without removal of duplicates (associations existing in both partitions). It is computed as $A_t = A_m + A_a$, where A_m is the total number of associations in the manual partition and A_a is the total number of associations in the automatic partition. We considered only associations from clusters representing three or more documents. It is easy to verify that this measure belongs to a [0,1] interval.

We also adopted *cluster recall* and *cluster precision* similarly to the measures of *recall* and *precision* typically used in information science research [23]. Rather than examining the number of relevant documents, we counted the number of correct associations. Therefore, we define cluster recall as:

$$\mathrm{CR} = \frac{A_c}{A_m},$$

where $A_c = A_a - E_i$ represents total number of correct associations in automatic partition. We define cluster precision:

$$\mathrm{CP} = \frac{A_c}{A_a}$$

It is easy to see that cluster recall reflects how well the clustering technique detects associations between documents and that cluster precision reflects how accurate the detected associations are.

Fig. 5 shows an example of manual partition (above) and automatic partition (below). In this example, the clustering algorithm made a mistake by placing document 5 with documents 1, 2, 3, 4 instead of 6, 7, 8. The incorrect associations are 5-1, 5-2, 5-3, 5-4. The missing associations are 5-6, 5-7, 5-8. The clustering error is $7/(1/2 \times 8 \times (8-1)) = 0.25$. The normalized clustering error is $7/(6+6+10+3) = 0.28$. Cluster recall is $(10+3-4)/(6+6) = 0.75$. Cluster precision is $(10+3-4)/(10+3) = 0.69$.

4.3. Research questions

In our research, we addressed the following questions.

Q1: Which technique produces output that requires less time for an expert to produce final clusters?

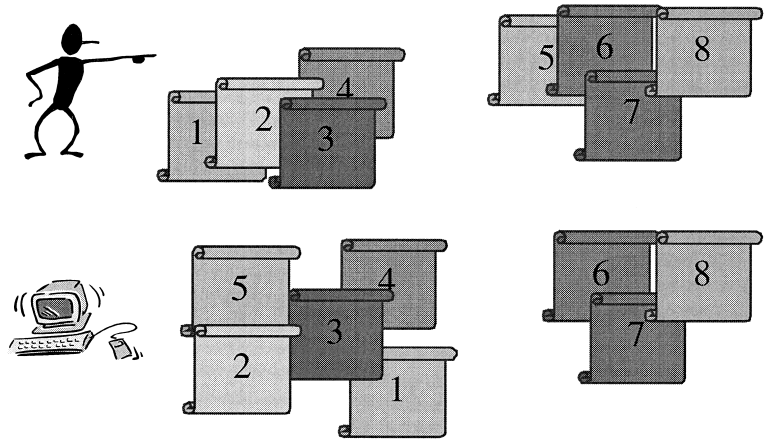


Fig. 5. The example of computing clustering error, recall, and precision.

Q2: Which technique provides greater cluster recall?
Q3: Which technique provides greater cluster precision?
Q4: Which technique provides smaller clustering error?
Q5: Which technique provides smaller normalized clustering error?

5. Results and discussion

It took 58 s to cluster the collection based on SOM on a DEC Alpha 3000/600 workstation (200 MHz, 128 MBs RAM). It took 34 s to cluster documents using Ward's technique on the same machine.

Below, we summarize our experimental findings. For all statistical significance tests, we used paired *t*-tests because two measures produced by the same subject were not independent. The only exception to this was time spent on the task, since we used only the first measurement from each subject. The data collected from the subjects. along with the statistical results are presented in Tables 2–6.

We randomized the order of tasks. In addition, in order to establish whether the order of tasks influenced the measures, we ran a regression of each measure on the order of tasks represented as a Boolean variable. We assigned 0 to this variable if Ward's clustering was processed by the subject first, and 1 if second. We did not find any statistically significant dependency. The minimum *p*-value for the coefficient in the regression was 0.20 in the case of normalized clustering error. This established that the influence of the order of tasks had been extremely weak and was not statistically significant. This was in agreement with the fact that most subjects did not even notice that the text messages were the same, but in the different order. Since the task was very laborious nobody appeared to have enough patience to verify that.

We have observed that the clusters created by subjects did not always share common words. This is

Table 2
Time spent on the tasks (in minutes)

Subject	SOM	Subject	Ward
1	45	1	10
2	90	2	10
3	20	3	15
4	60	4	90
5	50	5	60
6	29	6	60
7	27	7	60
8	90	8	35
		9	40
Average	51		42
Standard deviation	27		28
Confidence interval	31		18
Standard mean error	9.6		9.2

not surprising, since subjects clustered the comments based on their meaning but not the keyword representation as the algorithms did. For example, the comment ''Effective transmission of video over networks'' was placed by a subject into the same cluster with the comment ''bandwidth concerns — impact of remote collaboration'' presumably since both relate to networking issues. Those two comments do not have any common words, so they would never be placed together by our implementation of Ward's clustering and are quite unlikely to be so placed by SOM. This explains discrepancies between automatic and manual partitions.

Below, we present the results for each of the research questions.

Q1. The subjects spent less time correcting the Ward's clustering results than correcting the SOM results. However, the difference was statistically insignificant. The mean times spent on the task were 51 min for SOM and 42 min for Ward's clustering. The *p*-value was 0.19. Table 2 shows time spent on the task for all subjects. Since the variance in both groups was large, larger sample size appears to be necessary in order to establish statistical significance.

Table 3
Cluster recall

Subject	Ward	SOM
1	0.11	0.13
2	0.48	0.85
3	0.23	0.25
4	0.26	0.16
5	0.37	0.13
6	0.47	0.26
7	0.25	0.75
8	0.26	0.21
9	0.45	0.18
10	0.33	0.16
11	0.09	0.089
12	0.21	0.26
13	0.14	0.21
14	0.22	0.14
15	0.12	0.12
16	0.057	0.071
17	0.22	0.29
Average	0.254	0.254
Standard deviation	0.13	0.22
Confidence interval	0.063	0.10
Standard mean error	0.032	0.052

Table 4
Cluster precision

Subject	Ward	SOM
1	0.54	0.34
2	0.90	0.82
3	0.71	0.38
4	0.87	0.29
5	0.62	0.17
6	0.27	0.17
7	0.76	0.77
8	0.78	0.32
9	1.00	0.35
10	1.00	0.32
11	0.62	0.44
12	0.66	0.26
13	0.46	0.33
14	0.75	0.37
15	0.81	0.48
16	0.41	0.28
17	0.54	0.28
Average	0.70	0.38
Standard deviation	0.20	0.18
Confidence interval	0.097	0.084
Standard mean error	0.050	0.043

Q2. There was no statistical difference in cluster recall (Table 3) between the SOM results and the Ward's clustering results. The mean cluster recalls for both SOM and Ward's clustering were 0.25. The *p*-value was 0.30. Both techniques performed equally in detecting associations between documents. The 95% confidence interval for the difference in cluster recall was established at 0 ± 0.09. This is a rather wide interval so a larger sample size might establish a statistically significant difference.

Q3. Ward's clustering produced significantly better cluster precision (Table 4) than SOM. The mean cluster precisions were 0.38 for SOM, and 0.69 for Ward's clustering. The *p*-value for the paired *t*-test was 0.0014. The implication of this result is that Ward's clustering was more accurate in establishing associations between documents.

Q4. The SOM results exhibited significantly higher clustering error (Table 5) than the Ward's clustering results. The mean clustering errors were 0.080 for SOM and 0.051 for Ward's clustering. The *p*-value was 0.001. Table 3 shows the measurements for all subjects. However, since the clustering error measure favors a technique producing fewer clusters,

this is still not an ideal indication of quality. A more objective measure is considered in the next paragraph.

Q5. Ward's clustering produced a lower normalized clustering error (Table 6) than SOM. The mean normalized clustering errors were 0.71 for SOM, and 0.64 for Ward's clustering. The *p*-value for the paired *t*-test was 0.08. The implication is that, over all, the partition produced by Ward's clustering was closer to partitions produced by human experts.

The overall result is that on the EBS data set that we used Ward's clustering performed better. In addition, it produced a smaller number of associations, but was more accurate. The accuracy may be due to the ''shared keyword rule'' that we implemented, requiring the documents in a cluster to have at least one keyword in common. This conclusion applies only to the particular type of collection we used (electronic meeting messages) and may be sensitive to collection size. Since we are not aware of any evaluation study that was based on manual categorization of the output of automatic categorization and performed on a larger scale, we believe the results of our small scale experiments are valuable. This is also the first empirical study in the domain of text analysis of Kohonen self-organizing maps as a clustering tool.

Table 5
Clustering error

Subject	Ward	SOM
1	0.17	0.020
2	0.034	0.022
3	0.082	0.108
4	0.083	0.15
5	0.048	0.14
6	0.034	0.092
7	0.084	0.032
8	0.078	0.13
9	0.040	0.15
10	0.064	0.17
11	0.22	0.36
12	0.074	0.12
13	0.11	0.13
14	0.094	0.19
15	0.24	0.30
16	0.20	0.28
17	0.076	0.096
Average	0.104	0.160
Standard deviation	0.132	0.180
Confidence interval	0.032	0.044
Standard mean error	0.016	0.022

Table 6
Normalized clustering error

Subject	Ward	SOM
1	0.82	0.81
2	0.37	0.16
3	0.60	0.61
4	0.59	0.78
5	0.53	0.85
6	0.62	0.23
7	0.68	0.71
8	0.60	0.74
9	0.37	0.75
10	0.49	0.78
11	0.84	0.85
12	0.62	0.76
13	0.78	0.74
14	0.65	0.78
15	0.78	0.80
16	0.89	0.89
17	0.65	0.79
Average	0.64	0.71
Standard deviation	0.15	0.20
Confidence interval	0.072	0.096

6. Conclusion and future directions

We have concluded that our implementation of Ward's clustering is slightly more precise in detecting associations between documents, but that the performances of these techniques in terms of recall of those associations are not statistically different. This suggests that Kohonen's self-organizing map has clustering abilities close to those of known clustering techniques. Since the implementation of SOM for text analysis offers several additional valuable features such as providing labels for and visualizing proximity of the clusters [21], it may be a viable option for text clustering and categorizing systems.

Our research also resolved implementation issues related to two automatic text clustering techniques: Ward's clustering and Kohonen's Self-organizing Maps. These issues include:

Finding appropriate document representation for this task.

Adapting both techniques for creating non-overlapping partitions of text documents.

Resolving scalability issues by taking advantage of sparseness of representation in the domain.

In order to make the conclusions more general, experiments with different collections seem to be necessary and are under way. We are currently conducting a study involving Reuters collection [5], extensively used in text categorization research. We are planning to test clustering techniques embedded in an interactive search and visualization system.

Acknowledgements

Our research was supported by: Digital Library Initiative grant awarded by NSF/ARPA/NASA (''Building the Interspace: Digital Library Infrastructure for a University Engineering Community,'' PIs: B. Schatz, H. Chen et al., 1994–1998, IRI9411318); NSF/CISE grant (''Concept-based Categorization and Search on Internet: A Machine Learning, Parallel Computing Approach,'' PI: H. Chen, 1995–1998, IRI9525790.

References

[1] S.K. Card, G.G. Robertson, W. York, The WebBook and the Web Forager: An information workspace for the World-Wide Web, Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems, Vancouver, 1996, pp. 111–119.

[2] H. Chen, Artificial intelligence techniques for emerging information systems applications: trailblazing path to semantic interoperability, Journal of the American Society for Information Systems 49 (7) (1998) 579–581.

[3] H. Chen, C. Schuffels, R. Orwig, Internet categorization and search: a self-organizing approach, Journal of Visual Communication and Image Representation 7 (1) (1996) 88–102.

[4] D.R. Cutting, D.R. Karger, J.O. Pedersen, J.W. Tukey, Scatter/gather: A cluster-based approach to browsing large document collections, Proceedings of the 15th Annual International ACM Conference on Research and Development in Information Retrieval, 1992, pp. 318–329.

[5] S. Dumais, J. Platt, M. Sahami, D. Heckerman, Inductive Learning Algorithms and Representations for Text Categorization, 7th International Conference on Information and Knowledge Management, Bethesda, MD, 1998.

[6] A. El-Hamdouchi, P. Willett, Hierarchical document clustering using Ward's method, Proceedings of the 9th International Conference on Research and Development in Information Retrieval, Washington, DC, 1986, pp. 149–156.

[7] B.S. Everitt, Cluster Analysis, Wiley, New York, 1974.

[8] M.A. Hearst, J.O. Pedersen, Reexamining the cluster hypothesis: scatter/gather on retrieval results, Proceedings of the 19th Annual International ACM Conference on Research and Development in Information Retrieval, Zurich, 1996, pp. 76–84.

[9] M.A. Hearst, Interfaces for searching the Web, Scientific American, March (1997) pp. 68–72.

[10] T. Honkela, S. Kaski, K. Lagus, T. Kohonen, Newsgroup exploration with WEBSOM method and browsing interface, Report A32, Helsinki University of Technology, 1996.

[11] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[12] N. Jardine, C.J. van Rijsbergen, The use of hierarchic clustering in information retrieval, Information Storage and Retrieval 7 (1971) 217–240.

[13] T. Kohonen, Self-Organization and Associative Memory, Springer, 1989.

[14] T. Kohonen, Self-Organizing Maps, Springer, 1995.

[15] X. Lin, D. Soergel, G. Marchionini, A self-organizing semantic map for information retrieval, Proceedings of the 14th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, Chicago, IL, 1991, pp. 262–269.

[16] F. Murtagh, Multidimensional Clustering Algorithm, Physica-Verlag, Vienna, 1995.

[17] J.F. Nunamaker, A.R. Dennis, J.S. Valacich, D.R. Vogel, J.F. George, Electronic meeting systems to support group work: theory and practice at Arizona, Communications of the ACM 34 (7) (1991) 40–61.

[18] R.E. Orwig, H. Chen, J.F. Nunamaker, A graphical, self-organizing approach to classifying electronic meeting output, Journal of the American Society for Information Science 48 (2) (1997) 157–170.

[19] E. Rasmussen, Clustering algorithms, in: W.B. Frakes, R. Baeza-Yates (Eds.), Information Retrieval, Data Structures and Algorithms, Prentice-Hall, Englewood Cliffs, NJ, 1992, pp. 419–442.

[20] H. Ritter, T. Kohonen, Self-organizing semantic maps, Biological Cybernetics 61 (1989) 241–254.

[21] D. Roussinov, H. Chen, A scalable self-organizing map algorithm for textual classification: a neural network approach to thesaurus generation, Communication and Cognition — Artificial Intelligence 15 (1/2) (1998) 81–112.

[22] M. Sahami, S. Yusufali, Q.W. Baldonado, SONIA: A service for organizing networked information autonomously, Proceeding of the 3rd ACM International Conference on Digital Libraries, Pittsburgh, PA, 1998, pp. 237–246.

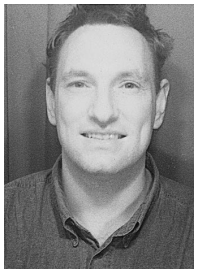
[23] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.

[24] B.R. Schatz, H. Chen, Building large-scale digital libraries, IEEE Computer 29 (5) (1996) 22–27.

[25] M.Q. Wang Baldonado, T. Winograd, SenseMaker: An information–exploration interface supporting the contextual evolution of a user's interests, Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems, Atlanta, GA, 1997, pp. 11–18.

[26] J. Ward, Hierarchical grouping to optimize an objection

function, Journal of the American Statistical Association 58 (1963) 236–244.
[27] O. Willet, Recent trends in hierarchical document clustering: a critical review, Information Processing and Management 24 (1988) 577–597.
[28] Y. Yang, C.G. Chute, An example-based mapping method for text categorization and retrieval, ACM Transaction on Information Systems 12 (3) (1994) 253–277.
[29] O. Zamir, O. Etzioni, O. Madani, R.M. Karp, Fast and intuitive clustering of Web documents, Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 1997, pp. 287–290.



Dmitri Roussinov is an Assistant Professor at the School of Information Studies, Syracuse University. His research interests, include human computer interaction, neural networks, and information retrieval. He has received an M.S. and B.S. in Computer Science from Moscow Institute of Physics and Technology, M.A. in Economics from Indiana University, and Ph.D. in Information Systems from the University of Arizona.

Dr. Hsinchun Chen is McClelland Professor of MIS and Andersen Professor of MIS at the University of Arizona, where he directs the UA/MIS Artificial Intelligence Group. Professor Chen is a Visiting Senior Research Scientist at NCSA. His articles have appeared in Communications of the ACM, IEEE Computer, Journal of the American Society for Information Science, IEEE Expert and many other publications. He has won numerous awards including Research Initiation (NSF), Best Paper (HICSS 1992), and an AT&T Foundation Award in Science and Engineering in 1994 and 1995. Professor Chen is a PI of the Illinois Digital Library Initiative project, and has received grant awards from the NSF, DARPA, NASA, NIH, NCSA, and NIJ. He is guest editor for special issues of IEEE Computer and the Journal of the American Society for Information Science.