Karagiannis & Reimer (Eds.): Practical Aspects of Knowledge Management.
LNAI 2569, pp. 168-178, ISBN 3-540-00314-2, ©Springer 2002.

Analysis of Clustering Algorithms for Web-based Search

Sven Meyer zu Eissen and Benno Stein
smze@upb.de, stein@upb.de

Paderborn University
Department of Computer Science
D-33095 Paderborn, Germany

Abstract Automatic document categorization plays a key role in the development of future interfaces for Web-based search. Clustering algorithms are considered as a technology that is capable of mastering this “ad-hoc” categorization task.

This paper presents results of a comprehensive analysis of clustering algorithms in connection with document categorization. The contributions relate to exemplar-based, hierarchical, and density-based clustering algorithms. In particular, we contrast ideal and real clustering settings and present runtime results that are based on efficient implementations of the investigated algorithms.

Key words: Document Categorization, Clustering, Clustering Quality Measures, Information Retrieval

1 Web-based Search and Clustering

The Internet provides a huge collection of documents, and its use as a source of information is obvious and became very popular. As pointed out and analyzed by Dennis et al. there is a plethora of Web search technology, which can broadly be classified into four categories [4]:

(1) *Unassisted Keyword Search.* One or more search terms are entered and the search engine returns a ranked list of document summaries. Representatives: Google (`www.google.com`) or AltaVista (`www.altavista.com`).
(2) *Assisted Keyword Search.* The search engine produces suggestions based on the user’s initial query. Representative: Vivisimo (`www.vivisimo.com`).
(3) *Directory-based Search.* Here, the information space is divided into a hierarchy of categories, where the user navigates from broad to specific classes. Representative: Yahoo! (`www.yahoo.com`).
(4) *Query-by-Example.* The user selects an interesting document snippet, which is then used as the basis of a new query.

In our working group we concentrate on developing smart interfaces for Web-based search. We think that the ideal search interface should model the search process within three phases: (a) An initialization phase according to the plain unassisted keyword search paradigm, (b) a categorization phase similar to the directory-based search paradigm, and (c) a refinement phase that may combine aspects from assisted keyword search and the query-by-example paradigm. Our realization of this process pursues a meta search strategy similar to that of Vivisimo; i. e., it employs existing search technology within the initialization phase.

The outlined ideal search process is the result of the following observations:

Existing search engines do an excellent and convenient job. They organize up to billions of documents which can be searched quickly for keywords, and, the plain keyword search forms the starting point for the majority of users. However, while this strategy works fine for the experienced human information miner, the typical user is faced either with an empty result list or with a list containing thousands of hits. The former situation is the result of misspelling or contradictory Boolean query formulation; it can be addressed with a syntactic analysis. The latter situation lacks a meaningful specification of context—it requires a semantic analysis, which can be provided by means of category narrowing. In this connection some search engines use a human-maintained predefined topic hierarchy with about 20 top-level categories like sports, art, music etc. Such static hierarchies are unsatisfactory within two respects: They require a considerable human maintenance effort, and, for special topics (example: “sound card driver”) the categories constitute an unnecessary browsing overhead which defers the search process. A powerful focusing assistance must be based onto a query-specific—say: ad-hoc—categorization of the delivered documents.

1.1 Contributions of the Paper

This paper focuses on ad-hoc categorization. Ad-hoc categorization comes along with two major challenges: Efficiency and nescience. Efficiency means that category formation must be performed at minimum detention, while nescience means that the category formation process is unsupervised: Except for experimental evaluation purposes, no predefined categorization scheme is given from which classification knowledge can be acquainted.

The paper in hand provides results of an analysis of clustering algorithms in connection with automatic document categorization. In particular, our contributions are threefold:

(1) The categorization performance of exemplar-based, hierarchical, and density-based clustering algorithms is shown within an idealized scenario. Such a scenario is characterized by the fact that no parameters of the clustering algorithm need to be estimated but the optimum values are chosen by a global analysis.
(2) In a realistic scenario, internal clustering quality measures are necessary to estimate cluster numbers, agglomeration thresholds, or neighborhood densities. From the various number of internal measures we have chosen approved ones and analyze the degradation of categorization performance compared to the optimum values.
(3) Several runtime issues are presented. They relate to both the algorithmic properties of the investigated algorithms and the difference when switching from an idealized to a realistic scenario.

Altogether, our analysis shall help to answer the question whether the investigated clustering technology is suited to master the pretentious job of ad-hoc categorization.

2 Document Representation, Clustering, and Quality Measures

The statistical method of variance analysis is used to verify whether a classification of given objects by means of nominal features is reflected in significant differences of depending metric features. Clustering can be considered as some kind of inverse operation: It tries to identify groups within an object set such that elements of different groups show significant differences with respect to their metric features.

Clustering algorithms operate on object similarities, which, in turn, are computed from abstract descriptions of the objects. Each such description is a vector $\mathbf{d}$ of numbers comprising values of essential object features. This section outlines the necessary concepts in connection with text documents: A suited object description, a related similarity measure, an overview of clustering algorithms, and—in particular, clustering quality measures for the analysis of an algorithm's categorization performance.

2.1 Document Representation

A common representation model for documents is the vector space model, where each document is represented in the term space, which roughly corresponds to the union of the m words that occur in a document collection [17, 11]. In this term space, common words are filtered out by means of a stop word list, words that are unique in the collection are omitted, and stemming is applied to reduce words towards a canonical form. The document collection $D = \{d_1, \ldots, d_n\}$ can then be described by means of vectors $\mathbf{d}_j = (w_{j1}, \ldots, w_{jm})$, where w_{ji} designates a weight of term t_i in document d_j. Widely accepted variants for the choice of w_{ji} are the following.

(1) The term frequency $tf(d_j, t_i)$ denotes the frequency of term i in document j. Defining the weights w_{ji} as $tf(d_j, t_i)$ implies that terms that are used more frequently are rated more important.

(2) The inverse document frequency is defined as $idf(t_i) := \log(\frac{n}{df(t_i)})$, where n is the total number of documents in the collection and $df(t_i)$ is the number of documents which contain the term t_i. The hypothesis is that terms that occur rarely in a document collection are of highly discriminative power. Defining $w_{ji} := tf(d_j, t_i) \cdot idf(t_i)$ combines the hypothesis with Point (1) and has shown to improve the retrieval performance [20]. Note that the representation of a single document requires knowledge of the whole collection if idf is used.

2.2 Document Similarity

Clustering exploits knowledge about the similarity among the objects to be clustered. The similarity φ of two documents, d_1, d_2, is computed as a function of the distance between the corresponding term vectors $\mathbf{d}_1$ and $\mathbf{d}_2$. There exist various measures for similarity computation, from which the cosine-measure proved to be the most successful for document comparison. It is defined as follows.

$$\varphi(d_1, d_2) = \frac{\langle \mathbf{d}_1, \mathbf{d}_2 \rangle}{||\mathbf{d}_1|| \cdot ||\mathbf{d}_2||},$$

where $\langle \mathbf{d}_1, \mathbf{d}_2 \rangle = \mathbf{d}_1^T \mathbf{d}_2$ denotes the scalar product, and $||\mathbf{d}||$ the Euclidean length. It calculates the cosine of the angle between two documents in $\mathbf{R}^m$. Note that a distance measure can easily be derived from φ by subtracting the similarity value from 1.

2.3 Clustering Algorithms

Let D be a set of objects. A clustering $\mathcal{C} = \{C \mid C \subseteq D\}$ of D is a division of D into sets for which the following conditions hold: $\bigcup_{C_i \in \mathcal{C}} C_i = D$, and $\forall C_i, C_j \in \mathcal{C} : C_i \cap C_{j \neq i} = \emptyset$.

Clustering algorithms, which generate a clustering $\mathcal{C}$, are distinguished with respect to their algorithmic properties. The following overview cannot be complete but outlines the most important classes along with the worst-case runtime behavior of prominent representatives. Again, n designates the number of documents in a given collection.

Iterative Algorithms. Iterative algorithms strive for a successive improvement of an existing clustering and can be further classified into exemplar-based and commutation-based approaches. These approaches need information with regard to the expected cluster number, k. Representatives: k-Means, k-Medoid, Kohonen, Fuzzy-k-Means [15, 9, 10, 24]. The runtime of these methods is $\mathcal{O}(nkl)$, where l designates the number of necessary iterations to achieve convergence.

Hierarchical Algorithms. Hierarchical algorithms create a tree of node subsets by successively merging (agglomerative approach) or subdividing (divisive approach) the objects. In order to obtain a unique clustering, a second step is necessary that prunes this tree at adequate places. Representatives: k-nearest-neighbor, linkage, Ward, or Mincut methods [6, 21, 7, 13, 23]. Usually, these methods construct a complete similarity graph, which results in $\mathcal{O}(n^2)$ runtime.

Density-based Algorithms. Density-based algorithms try to separate a similarity graph into subgraphs of high connectivity values. In the ideal case they can determine the cluster number k automatically and detect clusters of arbitrary shape and size. Representatives: DBSCAN, MAJORCLUST, CHAMELEON [22, 5, 8]. The runtime of these algorithms cannot be stated uniquely since it depends on diverse constraints. Typically, it is in magnitude of hierarchical algorithms, $\mathcal{O}(n^2)$, or higher.

Meta-Search Algorithms. Meta-search algorithms treat clustering as an optimization problem where a given goal criterion is to be minimized or maximized [1, 18, 19, 18]. Though this approach offers maximum flexibility, only less can be stated respecting its runtime.

2.4 Clustering Quality Measures

Many clustering algorithms do not return a definite clustering but a set of clusterings from which the best one has to be chosen. In particular, uniqueness within exemplar-based algorithms requires information about the cluster number, uniqueness within hierarchical algorithms requires an agglomeration threshold, or, within density-based algorithms, uniqueness requires a threshold for interpreting the neighborhood graph. If we had a measure to assess the quality of a clustering, the ambiguity could be mastered by simply computing several candidate clusterings and choosing the best one with respect to that measure. Note, however, that this is not a runtime problem in first place, but a problem of defining a suited quality measure.

Clustering quality measures evaluate the validity of a clustering and can be grouped into two categories: external and internal[1]. The following paragraphs introduce two clustering quality measures that are used within our experiments.

External Measures. External clustering quality measures use statistical tests to quantify how well a clustering matches the underlying structure of the data. In our context, the underlying structure is the known categorization of a document collection D as provided by a human editor. A broadly accepted external measure is the F-Measure, which combines the precision and recall ideas from information retrieval [12].

Let D represent the set of documents and let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a clustering of D. Moreover, let $\mathcal{C}^* = \{C_1^*, \ldots, C_l^*\}$ designate the human reference classification. Then the recall of cluster j with respect to class i, $rec(i,j)$, is defined as $|C_j \cap C_i^*|/|C_i^*|$. The precision of cluster j with respect to class i, $prec(i,j)$, is defined as $|C_j \cap C_i^*|/|C_j|$. The F-Measure combines both values as follows:

$$F_{i,j} = \frac{2 \cdot prec(i,j) \cdot rec(i,j)}{prec(i,j) + rec(i,j)}$$

Based on this formula, the overall F-Measure of a clustering is:

$$F = \sum_{i=1}^{l} \frac{|C_i^*|}{|V|} \cdot \max_{j=1,\ldots,k} \{F_{i,j}\}$$

A perfect clustering matches the given categories exactly and leads to an F-Measure value of 1.

Internal Measures. In absence of an external judgment, internal clustering quality measures must be used to quantify the validity of a clustering. Bezdek et al. present a thorough analysis of several internal measures, and, in this paper we rely on a measure from the Dunn Index family, which came off well in Bezdek et al.'s experiments [3, 2].

Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a clustering, $\delta : \mathcal{C} \times \mathcal{C} \to \mathbf{R}_0^+$ be a cluster-to-cluster distance measure, and $\Delta : \mathcal{C} \to \mathbf{R}_0^+$ be a cluster diameter measure. Then all measures $d : \mathcal{C} \to \mathbf{R}_0^+$ of the form

$$d(\mathcal{C}) = \frac{\min_{i \neq j}\{\delta(C_i, C_j)\}}{\max_{1 \leq l \leq k}\{\Delta(C_l)\}}$$

are called Dunn Indices. Of course there are numerous choices for δ and Δ, and Bezdek et al. experienced that the combination of

$$\delta(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} \psi(x,y) \quad \text{and} \quad \Delta(C_i) = 2\left(\frac{\sum_{x \in C_i} \psi(x, c_i)}{|C_i|}\right)$$

[1] Several authors also define relative clustering qualtity measures, which can be derived from internal measures by evaluating different clusterings and comparing their scores [9].

gave reliable results for several data sets from different domains. Here, ψ denotes a distance measure between the objects to be clustered, and c_i is the centroid of cluster C_i. Since we use the cosine similarity φ as similarity measure, we set $\psi = 1 - \varphi$.

Remarks. As mentioned at the outset, the use of external and internal measures corresponds to an idealized and realistic experimental scenario respectively: During ad-hoc categorization, only very little is known a-priori about the underlying structure of a document collection.

3 Experimental Setting and Results

The experiments have been conducted with samples of the Reuters-21578 text document database [14]. In this database a considerable part of the documents is assigned to more than one category. To uniquely measure the classification performance, only single-topic documents are considered within our samples comprising 1000 documents from exactly 10 classes each. To account for biased a-priory probabilities in the class distribution of Reuters-21578, the investigated test sets are constructed as uniformly distributed.

The generation of a sample requires some preprocessing effort that should not be underestimated. It includes the reading and parsing of the documents, the elimination of stop words according to standard stop word lists, the application of Porter's stemming algorithm [16], the computation of term frequencies, the creation of compressed index vectors, etc. Table 1 shows exemplary the runtime of important preprocessing steps, compression ratios, and term reduction ratios for different sample sizes.

Table 1. Runtime and impact of selected preprocessing steps, depending on the size of the investigated sample.

# Documents in sample	# Classes in sample	Indexing time	Compression time	Compression ratio	# Terms (raw)	# Terms (reduced)	Term reduction
400	10	1.80s	0.23s	98.6%	6010	4153	31%
800	10	2.88s	0.64s	99.0%	8370	5725	32%
1000	10	3.40s	0.89s	99.1%	9192	6277	32%

The plots at the end of this section present the results of the categorization performance experiments. In particular, the following three variates are combined:

(1) *Cluster Algorithm.* "k-Means" versus "Single-Link" versus "MAJORCLUST".

The algorithms are applied within a wide range of their respective parameters while paying attention to their special properties and strengths. More precisely: For k-Means all k-values between $1, \ldots, 20$ are considered. For Single-Link, clusterings at different agglomeration levels are considered. For MAJORCLUST, the threshold for edge weights is successively advanced within 20 steps, from 0 to 1.

The standard versions of Single-Link and MAJORCLUST operate on a completely connected distance or similarity graph. It is interesting—not only for experts in the

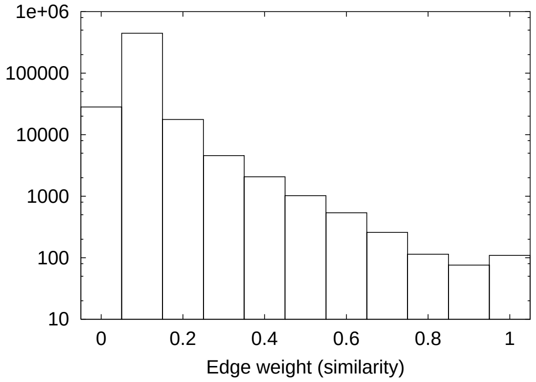

Figure 1. Distribution of edge weights in a completely connected graph with thousand nodes; nodes correspond to documents, edge weights correspond to similarities. Observe the logarithmic scale.

field of clustering—how these edge weights are distributed in our samples (cf. Figure 1). Of course, the creation of the graph imposes a severe performance burden, which can also be seen in the overview of Table 2.

(2) *Document Representation.* "tf" versus "$tf \cdot idf$".

The categorization performance of the three clustering algorithms is tested with both document representations.

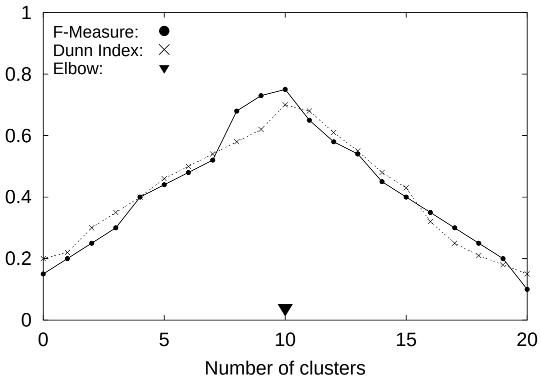

Figure 2. Fictitious curves of a consistent k-Means and perfect clustering quality measures: If the true number of classes is 10, and if the clustering algorithm behaves in a consistent manner, then the F-Measure values will follow the shape of a wedge with the maximum at $k = 10$. The values of a perfect Dunn Index will follow the F-Measure more or less, and the perfect elbow criterion indicates $k = 10$ as the optimum cluster number.

(3) *Scenario.* "Idealized" versus "Realistic".

In the idealized scenario, the best clustering of an algorithm is determined by means of the F-Measure. In the realistic scenario, the internal measures Dunn Index and variance drop (elbow criterion) are used to evaluate the clustering quality. To get an idea of the prediction quality, the variations in the F-Measure and the Dunn Index are plotted over the variation in selected parameters of the clustering algorithms. Remember that for both measures holds that larger values indicate better categorization performance. Figures 2 exemplifies a fictitious comparison for k-Means with variation in k, the cluster number parameter.

3.1 Categorization Results

The six plots in the Figures 3-5 show the categorization performance of k-Means, Single-Link, and MAJORCLUST (in this order). The plots on the left-hand side and right-hand side comprise the experiments with the document representation "tf" and "$tf \cdot idf$" respectively.

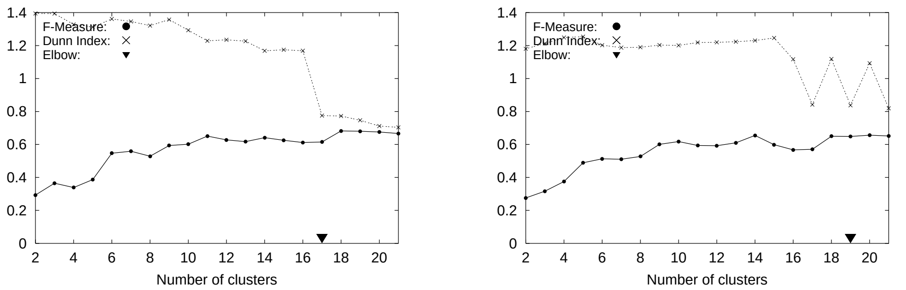

Figure 3. Clustering algorithm: k-Means with variation in the cluster number k and three random restarts for each k. Documents per sample: 1000. Classes per sample: 10. Document representation: tf (left) and $tf \cdot idf$ (right).

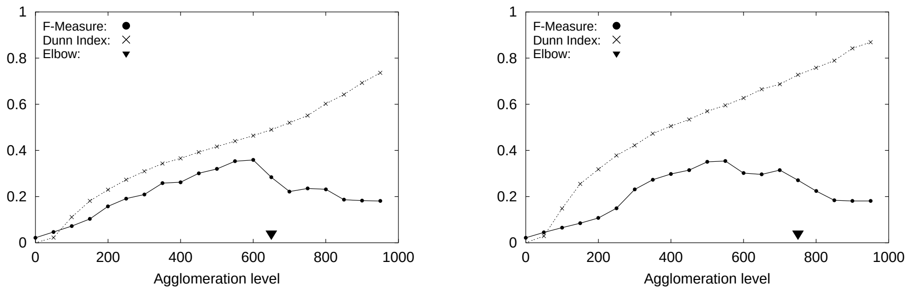

Figure 4. Clustering algorithm: Single-Link with variation in the agglomeration level (increment 50). Documents per sample: 1000. Classes per sample: 10. Document representation: tf (left) and $tf \cdot idf$ (right).

Table 2 comprises the key numbers with respect to categorization performance and runtime of the investigated clustering algorithms. The experiments were performed on a Pentium IV 1.7GHz. In this connection it should be noted that our text processing and classification environment is implemented in Java—but has been developed in the face of efficiency. Among others we developed tailored classes for symbol processing, efficient vector updating, and compressed term vectors.

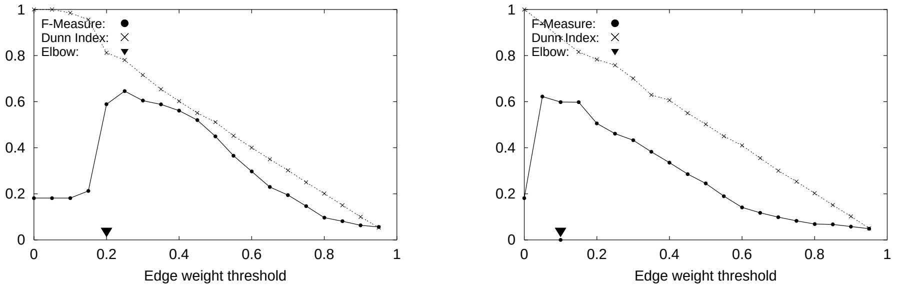

Figure 5. Clustering algorithm: MajorClust with variation in the edge weight threshold (increment 0.05) and three random restarts for each threshold. Documents per sample: 1000. Classes per sample: 10. Document representation: tf (left) and $tf \cdot idf$ (right).

4 Summary

In the long run, automatic text categorization will certainly become a part of standard Web search interfaces. However, each kind of such an ad-hoc categorization has to master two major challenges: Efficiency—category formation must be performed at minimum detention, and nescience—no predefined categorization scheme is given.

Clustering algorithms are considered as a technology that is capable of mastering the challenges, and this paper provides selected results of a comprehensive analysis. We compare the categorization performance of exemplar-based (k-Means), hierarchical (Single-Link), and density-based (MajorClust) clustering algorithms. The main result of the experiments can be comprised as follows.

Aside from the Single-Link algorithm, the categorization performance on samples (size: 1000, classes: 10) drawn from the Reuters-21578 text database achieves acceptable values—especially in an ideal scenario, where an external cluster performance measure is given. Even in a realistic scenario, reasonable F-Measure values can be realized. Here, a crucial role comes up to the internal clustering quality measure, which can completely ruin smart clustering technology. The presented results give an exam-

Table 2. Overview of some key numbers with respect to categorization performance and runtime of the investigated clustering algorithms. The first column corresponds to the ideal setting, column 2 and 3 to the realistic setting.

	F-**Measure Values**			**Runtime**		
	Maximum	according to Dunn Index	according to elbow criterion	Preprocessing	Graph creation	Clustering
k-Means	0.68	0.29	0.61	4.29s	–	2.33s
Single-Link	0.36	0.18	0.28	4.29s	5.78s	1.58s
MajorClust	0.65	0.18	0.58	4.29s	5.78s	2.38s

ple: The celebrated Dunn Index performs worse than a simple variance-based elbow criterion.

References

1. Thomas Bailey and John Cowles. Cluster Definition by the Optimization of Simple Measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 1983.
2. J. C. Bezdek, W. Q. Li, Y. Attikiouzel, and M. Windham. A Geometric Approach to Cluster Validity for Normal Mixtures. *Soft Computing 1*, September 1997.
3. J. C. Bezdek and N. R. Pal. Cluster Validation with Generalized Dunn's Indices. In N. Kasabov and G. Coghill, editors, *Proceedings of the 2nd international two-stream conference on ANNES*, pages 190–193, Piscataway, NJ, 1995. IEEE Press.
4. Simon Dennis, Peter Bruza, and Robert McArthur. Web searching: A process-oriented experimental study of three interactive search paradigms. *JASIST*, 53(2):120–133, 2002.
5. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD96)*, 1996.
6. K. Florek, J. Lukaszewiez, J. Perkal, H. Steinhaus, and S. Zubrzchi. Sur la liason et la division des points d'un ensemble fini. *Colloquium Methematicum*, 2, 1951.
7. S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32, 1967.
8. G. Karypis, E.-H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. Technical Report Paper No. 432, University of Minnesota, Minneapolis, 1999.
9. Leonard Kaufman and Peter J. Rousseuw. *Finding Groups in Data*. Wiley, 1990.
10. T. Kohonen. *Self Organization and Assoziative Memory*. Springer, 1990.
11. Gerald Kowalsky. *Information Retrieval Systems—Theory and Implementation*. Kluwer Academic, 1997.
12. Bjornar Larsen and Chinatsu Aone. Fast and Effective Text Mining Using Linear-time Document Clustering. In *Proceedings of the KDD-99 Workshop San Diego USA*, San Diego, CA, USA, 1999.
13. Thomas Lengauer. *Combinatorical algorithms for integrated circuit layout*. Applicable Theory in Computer Science. Teubner-Wiley, 1990.
14. David D. Lewis. Reuters-21578 Text Categorization Test Collection. `http://www.research.att.com/~lewis`, 1994.
15. J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
16. M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
17. C. J. van Rijsbergen. *Information Retrieval*. Buttersworth, London, 1979.
18. Tom Roxborough and Arunabha. Graph Clustering using Multiway Ratio Cut. In Stephen North, editor, *Graph Drawing*, Lecture Notes in Computer Science, Springer, 1996.

19. Reinhard Sablowski and Arne Frick. Automatic Graph Clustering. In Stephan North, editor, *Graph Drawing*, Lecture Notes in Computer Science, Springer, 1996.
20. G. Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1988.
21. P.H.A. Sneath. The application of computers to taxonomy. *J. Gen. Microbiol.*, 17, 1957.
22. Benno Stein and Oliver Niggemann. *25. Workshop on Graph Theory*, chapter On the Nature of Structure and its Identification. Lecture Notes on Computer Science, LNCS. Springer, Ascona, Italy, July 1999.
23. Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 1993.
24. J. T. Yan and P. Y. Hsiao. A fuzzy clustering algorithm for graph bisection. *Information Processing Letters*, 52, 1994.