Document Classification Methods for Organizing Explicit Knowledge

Heide Brücher, Gerhard Knolmayer, Marc-André Mittermayer

University of Bern
Institute of Information Systems
Research Group Information Engineering
Engehaldenstrasse 8
CH - 3012 Bern
Switzerland
http://www.ie.iwi.unibe.ch/

Phone +41 31 631 3809
Fax +41 31 631 4682
E-Mails: {bruecher|knolmayer|mittermayer}@ie.iwi.unibe.ch

Document Classification Methods for Organizing Explicit Knowledge

Summary

In this paper we describe the two classification approaches (i.e. categorization and clustering) and their preceding steps. For each approach we give a brief description of the underlying theory and outline the advantages and disadvantages of the different methods. Finally we specify potential application areas in accordance with knowledge management and illustrate exemplarily one topic in detail. We describe the enhancement of queries for illustration purposes because it is a common research problem with respect to information retrieval and, thus, to knowledge management.

1 Introduction

Today's organizations face a vast volume of knowledge and information. Most of the explicit knowledge is stored in different types of documents but only a few people (often only the authors of the documents) know where to locate them. There are plenty of ways to approach the problem of organizing knowledge in a company. This paper discusses advantages of document classification methods for organizing explicit knowledge. To simplify matters, documents are supposed to be plain text and tacit knowledge is not taken into consideration.

The objective of document classification is to reduce the detail and diversity of data and the resulting information overload by grouping similar documents together. The notion “document classification” is often used to subsume two types of analyses: document categorization and document clustering. The distinction is that categorization is a form of supervised and clustering an unsupervised approach of grouping textual objects.

The main classification objective, particularly with respect to knowledge management, is to simplify access to and processing of explicit knowledge. Classification supports analyzing the knowledge and, thus, can ease the

1. retrieval,
2. organization,
3. visualization,
4. development, and
5. exchange of knowledge [cf. BRU01].

For instance, knowledge abstraction is a means of the knowledge analyzing function and can be achieved by forming classes either by categorization or clustering. The use of identified class representatives reduces the amount of knowledge to cope with. Clustering documents - also a part of the analyzing function - helps to discover unknown structure in explicit knowledge. The discovered structure can be visualized and used to organize the knowledge.

Document categorization may be viewed as assigning documents or parts of documents in a predefined set of categories. Usually this set is created once and for all with so called training documents and, thus, remains unchanged over time. For applying document classification in general, some preprocessing tasks have to be executed; they are described in section 2. In section 3 we present a brief overview of some important document categorization methods and summarize results of surveys that have compared two to five classification methods. We also present new approaches for dealing with knowledge areas where a static set of categories seems inadequate.

In contrast to categorization, clustering is an unsupervised learning procedure. Cluster analyses are targeted on exploring similarities in the contents of the documents and arrange them in groups according to these properties. They are not based on a predefined structure of knowledge: Neither classes are predefined nor examples are given that show what types of relationships are expected between the objects. Any cluster analysis method requires some measures to be defined on the objects that have to be clustered and a threshold value indicating the (dis-) similarity between them. The objective is that each cluster is a collection of objects that are similar to each other within the same cluster and dissimilar to the members of other clusters. Clustering methods can be divided generally into hierarchical and partitional clustering methods. Within both types there exist several variants for defining the clusters. We discuss the most popular document clustering methods in section 4.

Finally we outline potential application areas and discuss exemplarily the enhancement of queries and cluster-based retrieval.

2 Document Preprocessing

Most of the methods used in document classification have been used in data mining applications. The data analyzed by data mining are numerical and, therefore, already in the format required by the algorithms.

To apply these algorithms for document classification one has to convert the words of the documents into numerical representations. This step is called document preprocessing and subsumes feature extraction, feature selection, and document representation as activities [WEI01].

2.1 Feature Extraction

Feature extraction is the first step in document preprocessing. The general problem in this phase is to generate a list of terms that describes the documents sufficiently [BOW01]. Therefore the training documents are parsed to determine a list of all words (i.e., features) contained in the documents. Afterwards feature reducing techniques are applied to reduce the dimension of the list created by the parsing process; this list is often denoted as dictionary. The most popular methods for this purpose are stop word removal and word stemming.

The main goal of stop word removal is to purge the dictionary from "noise" (e.g., articles, prepositions, numbers). It is usually realized by comparing the dictionary entries with a predefined stop word list and then eliminating accordances. Word stemming tries to treat terms that differ only in the affix (suffix or prefix), i.e., words with the same stem, as one single feature. Commonly applied word stemming techniques are affix removal, successor variety, and n-grams [BOW01].

2.2 Feature Selection

Feature extraction is followed by feature selection. The main objective of this phase is to eliminate those features that provide only few or less important information. This time statistical values are used to determine the most meaningful features. The most common indicators are term frequency (TF), inverse document frequency (IDF), and their multiplicative combination (TFxIDF).

By using TF it is assumed that important words occur more often in a document than unimportant ones. When applying IDF, the rarest words in the document collection are supposed to have the biggest explanatory power. With the combined procedure TFxIDF the two measures are aggregated into one variable. Whatever metric is used, at the end of the selection process only the top n words with the highest score are selected as features [WEI01].

2.3 Document Representation

Document representation is the final task in document preprocessing. Here the documents are represented in terms of those features to which the dictionary was reduced in the precedent steps. Thus, the representation of a document is a feature vector of n elements where n is the number of features remaining after finishing the selection process.

The whole document collection can therefore be seen as a mxn-feature matrix A (with m as the number of documents) where the element a_{ij} represents the frequency of occurrence of feature j in document i. Typical frequency measures are the above mentioned values TF, IDF, and TFxIDF; all positive values may be

replaced by 1, leading to a binary representation which indicates whether or not a certain feature appears in the document [WEI01].

3 Document Categorization

In general, document categorization only means assigning documents to a fixed set of categories. But in the domain of text mining document categorization also involves the preliminary process of automatically learning categorization patterns so that the categorization of new (uncategorized) documents is straightforward.

Major categorization approaches are decision trees, decision rules, k-nearest neighbors, Bayesian approaches, neural networks, regression-based methods, and vector-based methods. In section 3.1 we briefly describe these methods and discuss their relative merits. We compose issues of other studies that compared some of these algorithms among each other (section 3.2). Finally we illustrate an extension of categorization approaches which tries to overcome the problem of static categories (section 3.3).

3.1 Categorization Methods

3.1.1 Decision Trees

Decision tree methods rebuild the manual categorization of the training documents by constructing well-defined true/false-queries in the form of a tree structure where the nodes represent questions and the leafs the corresponding category of documents [GER01]. After having created the tree, a new document can easily be categorized by putting it in the root node of the tree and let it run through the query structure until it reaches a certain leaf.

The main advantage of decision trees is the fact that the output tree is easy to interpret even for persons who are not familiar with the details of the model. The tree structure generated by the model provides the user with a consolidated view of the categorization logic and is therefore a useful information.

A risk of the application of tree methods is known as "overfitting": A tree overfits the training data if there exists an alternative tree that categorizes the training data worse but would categorize the documents to be categorized later better [GRE01]. This circumstance is the result of the algorithm's intention to construct a tree that categorizes every training document correctly; however, this tree may not be necessarily well suited for other documents. This problem is typically moderated by using a validation data set for which the tree has to perform in a similar way as on the set of training data.

Other techniques to prevent the algorithm from building huge trees (that anyway only map the training data correctly) are to set parameters like the maximum depth of the tree or the minimum number of observations in a leaf. If this is done, Decision Trees show very good performance even for categorization problems with a very large number of entries in the dictionary [GER01].

3.1.2 Decision Rules

Decision rule algorithms construct for every category a rule set that describes the profile of this category. In general, a single rule consists of a category name and a feature of the dictionary which is typical for the training documents belonging to the considered category. Then the rule set is created by combining the separate rules with the logical operator "or". Usually not all of the rules are required to categorize the documents adequately. Therefore, heuristics are applied to reduce the size of the rule sets. The goal is to achieve a reduced rule set per category which, however, does not affect the categorization of the training documents [APT94].

The main advantage of decision rules is the possibility to create local dictionaries (i.e., per category) during the feature extraction phase [APT94]. Consider the homonym "bark", describing as well the outer part of a tree and the sounds of a dog. In an universal dictionary "bark" will be listed only once, thus documents with "the bark" and documents with "to bark" get a similar feature vector even they have nothing in common. Local dictionaries are - to a certain extent - able to distinguish different meanings of homonyms. If in the precedent example dog-related documents belong to a different category than tree-related do, the local dictionaries of these categories both list the term "bark" but with different meaning, because once it appears in a rule set with "dog" and another time with "tree".

A disadvantage is that it is impossible to assign a document exclusively to one category because rules from different categories are applicable. In this case the document is usually assigned to several categories.

3.1.3 k-Nearest Neighbor

Whereas the categorization methods described above are based on a learning phase in which the guiding principles to be used are determined and consist therefore of at least two phases, k-nearest neighbor completely skips the learning phase and categorizes on-the-fly.

The categorization itself is usually performed by comparing the category frequencies of the k nearest documents (neighbors). How close documents are to each other can be evaluated by measuring for instance the angle between the two feature vectors or calculating the Euclidean distance between the vectors. In the latter case the feature vectors have to be normalized to length 1 to take into account that the size of the documents (and, thus, the length of the feature vectors) may differ.

A doubtless advantage of the k-nearest neighbor method is its simplicity. It has reasonable similarity measures and does not need any resources for training. K-nearest neighbor performs well even if the category-specific documents form more than one cluster because the category contains, e.g., more than one topic [GER01]. This situation is badly suited for most categorization algorithms.

A disadvantage is the above-average categorization time because no preliminary investment (in the sense of a learning phase) has been done. Furthermore, with different numbers of training documents per category the risk increases that too many documents from a comparatively large category appear under the k nearest neighbors and thus lead to an inadequate categorization.

3.1.4 Bayesian Approaches

A comprehensive explanation of Bayesian categorization approaches is quite challenging. Therefore, we do not describe computational details of these methods.

There are two groups of Bayesian approaches in document categorization: Naïve and non-naïve Bayesian approaches. The naïve part of the former is the assumption of word (i.e. feature) independence, meaning that the word order is irrelevant and consequently that the presence of one word does not affect the presence or absence of another one. This assumption makes the computation of Bayesian approaches more efficient. But although the assumption is obviously severely violated in every language, it has been shown that the classification accuracy is not seriously affected by this kind of violations [DOM97]. Nevertheless, several non-naïve Bayesian approaches eliminate this assumption [cf. LAM97].

Naïve Bayesian approaches have been developed comparatively early and have been studied frequently in data mining before the topic of document categorization gained importance. They perform as well as newer, more sophisticated methods [WIT00] and also show a very good runtime-behavior during the categorization of new documents [HER01]. A disadvantage of Bayesian approaches in general is that they can only process binary feature vectors [LAM97] and, thus, have to abandon possibly relevant information.

3.1.5 Neural Networks

Different neural network approaches have been applied to document categorization problems. While some of them use the simplest form of neural networks, known as perceptrons, which consist only of an input and an output layer [NGG97], others build more sophisticated neural networks with a hidden layer between the two others [RUI98]. In general, these feed-forward-nets consist of at least three layers (one input, one output, and at least one hidden layer) and use backpropagation as learning mechanism [KRA98]. However, the comparatively old perceptron approaches perform surprisingly well [NGG97].

The advantage of neural networks is that they can handle noisy or contradictory data very well [KRA98]. Furthermore some types of neural networks are able to comprehend fuzzy logic [NIE95], but one has to change from backpropagation as learning mechanism to counterpropagation (for which worse categorization results are reported [RUI98]). The advantage of the high flexibility of neural networks entails the disadvantage of very high computing costs. Another disadvantage is that neural networks are extremely difficult to understand for an average user; this may negatively influence the acceptance of these methods.

3.1.6 Regression-based Methods

For this method the training data are represented as a pair of input/output matrices where the input matrix is identical to our feature matrix A and the output matrix B consists of flags indicating the category membership of the corresponding document in matrix A. Thus B has the same number of rows like A (namely m) and c columns where c represents the total number of categories defined. The goal of the method is to find a matrix F that transforms A into B' (by simply computing B'=A*F) so that B' matches B as well as possible. The matrix F is determined by applying multivariate regression techniques [YAN94].

An advantage of this method is that morphological preprocessing (e.g., word stemming) of the documents can be avoided without loosing categorization quality [YAN94]. Thus, regression-based approaches become truly language-independent. Another advantage is that these methods can easily be used for both single-category and multiple-category problems. Unfortunately regression-based methods are not very popular in the categorization community and, therefore, investigations comparing regression-based methods with others are relatively rare.

3.1.7 Vector-based Methods

We discuss two types of vector-based methods: The centroid algorithm and support vector machines [JOA98].

One of the simplest categorization methods is the centroid algorithm. During the learning stage only the average feature vector for each category is calculated and set as centroid-vector for the category [MAD01]. A new document is easily categorized by finding the centroid-vector closest to its feature vector. The distance can be measured as described in section 3.1.3.

Unless the document clusters overlap each other, this method does not need many training documents. If, however, the document clusters overlap each other or the category consists of two or more different topics (clusters), the algorithm performs often poor. The method is also inappropriate if the number of categories is very large [GER01].

Support vector machines (SVM) need in addition to positive training documents also a certain number of negative training documents which are untypical for the category considered. SVM is then looking for the decision surface that best separates the positive from the negative examples in the n-dimensional space. The document representatives closest to the decision surface are called support vectors. The result of the algorithm remains unchanged if documents that do not belong to the support vectors are removed from the set of training data. An advantage of SVM is its superior runtime-behavior during the categorization of new documents because only one dot product per new document has to be computed. A disadvantage is the fact that a document could be assigned to several categories because the similarity is typically calculated individually for each category. Nevertheless, SVM is a very powerful method and has outperformed other methods in several studies [cf. DUM98, HEA98, JOA98, YAN99, SIO00].

3.2 Comparison of Categorization Methods

As shown, many algorithms have been proposed for document categorization. Some papers compare the effectiveness of selected algorithms. Table 1 provides an overview on this work. The surveys use manually pre-categorized documents as input data set. Usually this data set is split in a training and a test part; the latter is needed to determine the quality of the algorithm developed. Commonly used test collections are the Reuters collection (newswires from Reuters; downloadable for instance at http://www.research.att.com/~lewis/reuters21578.html) and the OHSUMED collection (abstracts from medical journals; downloadable for instance at http://trec.nist.gov/data.html).

The comparison of algorithms and heuristics is scientifically demanding [cf., e.g., GOL85]. The results may be heavily dependent of the test data set. Furthermore, several parameters usually have to be defined to initialize the procedures and the performance may depend on their initialization. If various “standard data sets” exist, the heuristics may even be tuned to deliver high efficiency for these data sets.

Taken these limitations into consideration, we have to emphasize that the SVM method has outperformed the other methods in several comparisons. Furthermore, there are results indicating that combinations of basics methods often provide better results than the application of the underlying “pure” methods.

Authors	Type of methods investigated								Test corpus			Main results
	Dec. Tree	Dec. Rules	k-NN	Bayes. Appr.	Neural Netw.	Regr.-based	Centroid	SVM	REUTERS	OHSUMED	Others	
[APT94]	x	x		x					x		x	Swap-1 (Dec. Rule) shows the best performance, Bayesian Independence Classifier and Decision Tree perform similar but worse.
[LEW94]	x			x					x		x	Both algorithms perform similar.
[LAR96]			x	x			x				x	The algorithms perform similar but the combination of the algorithms yields better results.
[DUM98]	x			x			x	x	x			Support Vector Machines delivers the best performance, Find Similar (Centroid) the worst one.
[JOA98]	x		x	x			x	x	x			Support Vector Machines has the best performance.
[LAM98]			x				x		x	x		Combinations of ExpNet (k-NN) and Rocchio (Centroid) or ExpNet and Widrow-Hoff (Centroid) perform better than the basic algorithms.
[RUI98]					x		x			x		Neural Networks perform better than Rocchio (Centroid).
[YAN99]			x	x	x	x		x	x	x		With few documents per category (< 10), Support Vector Machines, k-NN, and LLSF (regression-based) perform significantly better than the other methods; however, with more than 300 documents per category all the methods perform similarly.
[SIO00]			x					x			x	Support Vector Machines performs better than k-NN.

Table 1: Comparison of Categorization Methods

3.3 Extensions of Categorization Methods

All methods described in section 3.1 assume that the underlying set of categories is static over time. They do not provide mechanisms for the circumstance that the category structure might change because, e.g., a new category comes up (category discovery) or some of the categories merge due to lack of new documents. And since categories defined earlier cannot capture sufficiently the characteristics of documents currently held, we should note that the methods discussed pay no attention to the dynamics of today's business.

Unfortunately, previous research focused predominantly on improving the accuracy of the categorization process. Thus, a scientific effort is required to develop extensions for the established categorization methods which are able to handle dynamics.

To date only a few studies with the objective of developing evolutionary extensions to existing categorization methods have been carried out. A promising approach is the mining-based category evolution technique called MiCE [WEI01]. The MiCE technique features two main operations: category decomposition and category merging. A category may be decomposed if the topic covered by a subset of documents that are similar to each other differs from the one covered by the remaining documents in the category. On the other hand the goal of category merging is to pool two or more categories if they cover similar topics [WEI01]. A disadvantage of MiCE is the fact that, at most, one category is assigned to each document.

4 Document Clustering

Clustering deals with m objects that are described by n features [GUH98]. The features for each object can be regarded as vectors used to position the object in the n dimensional space formed by the features. The objects are assigned to different clusters.

Several cluster analysis methods have been developed. The tree shown in Figure 1 provides an overview over the different types of methods.

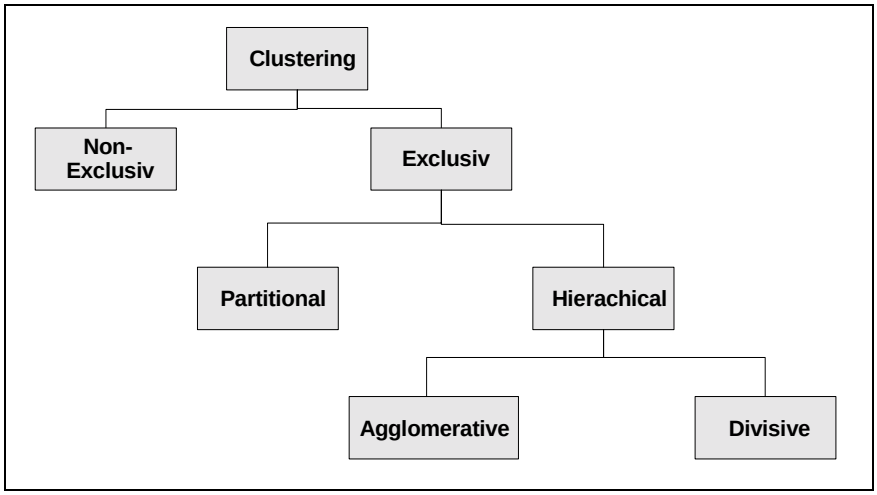

Figure 1: Types of Clustering Methods

The first distinction made is whether a clustering method produces exclusive or non-exclusive clusters [ZAH71]. Exclusive means the clusters formed are disjunctive, i.e., the object can belong only to one specific cluster. Even if an object could be assigned to two clusters because of ties in the measurement criterion values, the clustering method has to determine to which group the object should belong. A few cluster analysis methods allow objects to be member of more than one cluster, resulting in overlapping clusters (non-exclusive clustering) [SHE79; JAR69].

The mostly recommended clustering methods are hierarchical and partitional clustering algorithms. Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones or by splitting larger clusters. In section 4.1 hierarchical clustering methods will be described in more detail. Partitional clustering results in clustering the objects into a predefined number of clusters; it will be described in section 4.2 in more detail. We also discuss briefly aspects of proximity analysis because the choices of the clustering method and the clustering criterion determine the way in which the proximity between clusters is measured.

4.1 Hierarchical Clustering

The result of a hierarchical clustering process is a sequence of nested partitions. These are derived from a set of objects by using a certain similarity criterion [AND73]. Most hierarchical methods store the distances between the objects in a matrix D of dimension mxm. At any stage of the procedure, a hierarchical clustering technique either merges clusters (agglomerative methods) or splits them (divisive methods). This results conceptually in a tree-like structure. The clusters of objects formed at any stage are non-overlapping (mutually exclusive). The results

of these methods can be displayed in a dendrogram, a tree-like diagram which depicts the mergers or divisions made at successive level. The dendrogram provides the opportunity to choose a certain tree-level and, thus, a certain distance level where the grouping occurred. Consequently the number of clusters has not to be defined in advance, but can be determined with a-posteriori knowledge.

An agglomerative algorithm starts with disjoint clustering, placing each of the n objects in an individual cluster. It successively combines pairs of clusters until finally reaching the point where all objects form one large cluster. In each of the subsequent steps, the two closest clusters will merge. The others remain unchanged [JAI88]. Before the cluster analysis can proceed, the proximities for the newly formed cluster must be calculated in relation to the unchanged clusters. A divisive algorithm performs the tasks in reverse order [JOH98; MIR96].

Most hierarchical clustering algorithms are variants of four basic algorithms: Single-link, Complete-Link, and Average-Link are agglomerative methods and Ward's algorithm [WAR63] exists as an agglomerative as well as a divisive version [AND73]. All these methods use different measurements for determining the proximity between two clusters. The agglomerative methods have in common that whatever proximity measurement is chosen, the solution continues with sequent levels always merging the two nearest clusters until reaching the last level of the hierarchy [JAI88]. The divisive methods split the "worst" cluster. The criterion used to determine the "worst" cluster may be the number of objects in the cluster (thus splitting the largest cluster) or the cluster with the largest variance or the largest sum-squared-error to name just a few. The computations required by divisive clustering are more intensive than for agglomerative clustering methods and, thus, agglomerative approaches are more common.

The Single-Link, Complete-Link, and Average-Link algorithms differ in the way they compute the similarity between a pair of clusters. In the Single-Link method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters. This procedure will string objects together to form clusters and the resulting clusters tend to appear as straggly, elongated chains.

In the Complete-Link Method, the distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors") [VRY77]. Those two clusters are merged for which the distance between their furthest objects is minimal. This method usually performs quite well in cases where the objects naturally form distinct clumps. If the clusters tend to be somehow elongated or of a chain-type, then this method is inappropriate. The clusters obtained by the Complete-Link algorithm are more compact than those obtained by the Single-Link algorithm. The Single-Link algorithm is more versatile than the Complete-Link algorithm. In many applications the Complete-Link algorithm resulted in more useful results than the Single-Link algorithm [JAI88].

In the Average-Link Method the distance between two clusters is computed as the average of the distances between all points in these clusters. This approach can be regarded as a compromise between the Single- and the Complete-Link Method. Furthermore it is less sensitive to outliers. This means that if an object is quite distinct from the other ones – i.e., lies far away from the cluster centroid - this is not likely to skew the clustering result.

Ward's method uses an analysis of variance approach to evaluate the distances between clusters. Cluster membership is assessed by calculating the total sum of squared deviations from the mean of a cluster. The criterion for fusion is that it should produce the smallest possible increase in the error sum of squares [WAR63]. This involves finding the mean of each cluster and the distance to each object contained in each cluster, then squaring these distances and summing the squared distances for all the objects in all clusters. In general, this method is regarded as very efficient, however, it tends to create clusters of small size and thus constructs a highly diversified dendrogram [JAI99].

The low calculation effort of hierarchical clustering methods resulting from the effect that each merger respectively split-up of clusters restricts the number of alternatives to be considered is often mentioned as advantage. A disadvantage of hierarchical clustering methods is that they is not reversible: Each step is definitive and cannot be reconsidered later. A further disadvantage is that the results are not robust: They depend on the initialization of the cluster algorithm and does not protect against outliers; results can be influenced heavily by extraneous data or noise in the set of objects.

4.2 Partitional Clustering

The basic concept underlying partitional clustering is the decomposition of a set of objects into k disjoint, flat clusters without imposing a hierarchical structure. Instead of a clustering structure, such as the dendrogram obtained by the hierarchical techniques, the partitional clustering algorithms result in a unique partition of the objects. Whereas hierarchical clustering uses a proximity matrix to determine relationships between objects, partitional clustering uses a feature vector matrix. Instead of representing the difference between each object as a distance, one can use the feature matrix A for comparisons. The features of each object are compared and the object is placed in a cluster with other objects of similar patterns.

Before explaining different versions of this group of methods some general properties are mentioned. Partitional methods are well suited for large object sets for which the construction of a dendrogram is computationally prohibitive. The choice of the number of desired output clusters is a key problem arising from using a partitional algorithm [JAI99]. Some methods of resolution on this key design decision have been proposed [DUB87].

Different partitional clustering methods are briefly described below: The deterministic partitional clustering methods K-Means clustering, Single-pass clustering, and Nearest Neighbor clustering and the probabilistic partitional clustering approach Expectation Maximization.

The K-Means-Method belongs to a family of iterative partitional clustering methods that repeat the algorithm until cluster membership stabilizes. First the number k of clusters to be formed has to be defined. Then, k cluster centers are chosen, either arbitrarily or according to prior knowledge about the objects being classified [JAI99]. The first step is to assign each object to be clustered to its nearest cluster center using the Euclidean distance metric. In a second step the cluster centers are computed for the formed clusters. These cluster centers are used as new centroids of the clusters. Based on the calculated cluster centroids the first step is repeated and the objects to be clustered are again assigned to their nearest cluster center. And then the second step is performed once again. These two steps are repeated until there is no change and the cluster centers have stabilized and remain widely constant.

The K-Mean-Method determines a set of k clusters that minimizes the squared-error criterion. The squared error for a cluster is the sum of the squared Euclidean distances between each element in the cluster and the cluster centroid. The K-means algorithm reduces the ‘distances’ between the points in each cluster in each iteration of the algorithm. The treatment of outliers is poor. Stop criteria can be based on the sum of squared errors, or on a predetermined number of iterations, or as mentioned before on unchanged assignments in successive iterations.

The Single-Pass Algorithm starts with an initial set of empty clusters. It picks a document at random or in a predefined order. The picked document is treated as a new cluster with only one member and compared with all other clusters. If the linkage between the new cluster and any other cluster is above a usually predetermined threshold, then the new cluster is merged with the closest cluster. Otherwise a new cluster with only one member is added to the cluster set. These two steps are repeated until all the documents are assigned.

The Nearest Neighbor clustering method is also iterative and similar to the hierarchical Single Link method. It uses the nearest distance as a threshold to determine whether objects will be added to existing clusters or a new cluster is created. The similarity respectively distance measurement between joint attributes of the objects is based on Euclidean distance measurements [ZHO01].

The basic idea behind probabilistic clustering is to assign probabilities for the membership of an object to a cluster. The Expectation Maximization algorithm [DEM77] is a well-known technique for estimating the missing feature parameters [JAI99]. It calculates the maximum likelihood of the parameters of an underlying distribution from the given data set when the data set is incomplete or has missing values. Sometimes the missing values are due to limitations of the observation process. In other cases the likelihood function is analytically intractable but it can

be simplified by assuming the existence of additional but missing (i.e. hidden) parameters. Expectation Maximization as well as K-Means tend to converge to one of several local minima. It is especially sensitive to the chosen initial conditions.

4.3 Fuzzy Clustering

Fuzzy clustering is a non-exclusive clustering method generating a given number of partitions with fuzzy boundaries. Each object can belong to more than one cluster. Hence, the fuzzy clusters are not necessarily disjoint [JAI99]. The basic idea of fuzzy clustering is to represent the vagueness in everyday life [ZAD65; ZHO01]. Often situations occur in which some data points respectively objects are located between several cluster centroids. With fuzzy clustering each object belongs to all clusters simultaneously, but to different degrees.

Most fuzzy clustering algorithms try to determine an optimal clustering by minimizing an objective function [HOE97]. Each cluster is represented by a cluster prototype. This prototype consists of a cluster center and maybe some additional information about the size and the shape of the cluster. The cluster center is an instantiation of the features used to describe the domain, just as the objects in the object set. However, the cluster center is computed by the clustering algorithm and may or may not appear in the object set.

The ambiguous situation of non-exclusive cluster membership can be described mathematically by a fuzzy membership function. The membership function calculates a membership vector for each object. The i-th element of the membership vector – so called membership factor f_i - of an object indicates the membership degree of the object to the i-th cluster. In particular the degrees of membership to which a given object belongs to the different clusters are computed from the distances of the object to the cluster centers. The closer a data point lies to the center of a cluster the higher is its degree of membership to this cluster [HOE97]. Larger membership factor values indicate higher confidence in the assignment of the object to the cluster [JAI99]. The task is to minimize the distances of the objects to the cluster centers in order to maximize the degrees of membership.

The most widely known fuzzy clustering algorithm is the Fuzzy C-Means[1] algorithm [KRI94]. The algorithm uses a random initialization of the cluster center and improves the objective function iteratively. The distance is measured by the Euclidean metric. According to the specified number of classes, the Fuzzy C-Means algorithm shows the tendency to partition the objects in clusters of hyperspherical shape with very similar numbers of objects. Although it performs better than the exclusive k-means algorithm at avoiding local minima, Fuzzy C-Means

[1] The C in Fuzzy C-Means symbolizes the number of clusters to be built like the K in the name of K-Means algorithm.

may still converge to a local minimum of the squared error sum [JAI99]. Furthermore, the initialization of the cluster centers at the start of the algorithm may influence the output of the clustering algorithm remarkably. One way to reduce this influence may be to repeat the clustering several times with different initializations and select the best partition by comparing the results of the objective function. An advantage is that fuzzy clustering can deal with overlapping cluster boundaries.

4.4 Comparison of Document Clustering Methods

Most document clustering approaches are based either on distance and similarity measures or on probabilistic methods [FRA99]. Distance-based methods such as K-Means, hierarchical and nearest neighbor clustering as well as probabilistic clustering methods use a selected set of words appearing in different documents as features. Each document is represented by a feature vector and can be viewed as a point in a multi-dimensional space [JAI99].

Clustering documents in a multi-dimensional space using distance or probabilistic based clustering methods is quite complex due to the fact that the distance or probabilistic measurement has to master this dimension. Feature vectors must be scaled to avoid skewing the result by different document lengths or possibly by the occurrence of different words in the documents [JAI99].

Probability-based approaches typically assume that all elements of the cluster or document representative (vector) are statistically independent of each other. This assumption is not realistic. Often positive correlations between features exist which skew the clustering results.

One overall problem is that the interpretation of the clusters may be difficult. Most clustering algorithms prefer certain cluster shapes and the algorithms will always assign the data to clusters of such shapes even if there exist no clusters in the data. Therefore, if the goal is not just to compress the data set but also to make inferences about its cluster structure, it is essential to determine whether the data set indicates a clustering tendency [VRI79].

Quite different results may be obtained when the number of clusters is changed. Hierarchical methods elude this problem: The dendrogram allows the analyst to choose different cluster levels and, thus, to determine the number of clusters used in subsequent applications. Good initialization of the cluster centroids may also be crucial for instance for the K-Means or the Fuzzy C-Means clustering method; some clusters may be left empty if their initially chosen centroids lie far from the distribution of objects.

Furthermore, the choice of a clustering method or clustering criterion will determine the way in which the proximity between two clusters is measured. For example, using the Single-Link method, the proximity between two clusters is the

highest similarity (or smallest distance) between any two objects from the different clusters. These are the nearest neighbors. By contrast, with the Complete-Link method, the similarity between two clusters is the largest distance between any two objects in the different clusters.

Finally, clustering depends on the application and the task to be solved. Thus, the same set of objects often needs to be partitioned differently for diverse purposes [JAI99].

5 Potential Application Areas

The objective of document classification in general is to find similar documents and group them together. The more frequently two terms occur in the same text, the more likely they are about the same concept. Thus classification provides groupings that can be used to speed up searches, to provide assistance in interpreting and analyzing the information contained in relevant classes and documents, to automatically formulate queries for a subsequent search of similar documents, and to construct user profiles [BRU01].

Potential application areas are shown in Table 2 [cf. DOE01; GER01].

Application Example	Knowledge Management Functions supported
Enhancement of queries (e.g., of search engines on the Web)	Retrieval
Clean-up of document collections (e.g., on a file server)	Organization
Reorganization and analysis of databases (e.g., patent documents)	Organization
E-Mail filtering and sorting	Organization
Analysis of news tickers (e.g., Reuters)	Visualization
Stock quote forecasting	Development
Automatic message forwarding (e.g., at help desks)	Exchange

Table 2: Potential Application Areas

In the following we discuss exemplarily the enhancement of queries with clustering methods to give an insight into the potential of classification methods with respect to a certain problem.

Typically a user expresses his information need in the form of a request, the query. User queries can range from multi-sentence full descriptions of an information

need to a few words. The vast majority of retrieval systems currently in use range from simple Boolean systems to systems using statistical or natural language processing. The results of queries are usually presented in a hit list with the most relevant documents on top. For instance, as relevance criteria the following rules (for instance) may be applied [cf. MYA01]:

- The more terms of the query appear in a document the more probable this document is relevant.
- The more terms of the query appear in the meta data of a document the more probable this document is relevant.
- The more terms of the query appear as bold text in a document the more probable the document is relevant.
- The more a certain term of the query appears in a document the more probable this document is relevant.
- Documents that contain rarely used search terms are probably more relevant than documents containing frequently used search terms.
- A small document containing all terms of the query is probably more relevant than a larger one.
- The closer the search terms appear in a document the more probable this document is relevant.
- The earlier a certain term of the query appears in a document the more probable this document is relevant.

Web Search Engines typically use such kind of relevance criteria to rank their results in order to prevent the user from having to scroll through all the hits to find pages of interest. However, we all know that the output obtained is not always satisfying. Therefore other approaches group similar documents or documents belonging to the same topic together and thus give the user faster access to the documents of interest. Several human-edited directories exist in the Web (e.g., Yahoo!, Open Directory Project) that automatically sort the search results by the categories defined on the site. Although this idea is a good one, it provides only one single categorization, the view of the site provider. More helpful would be solutions in which the categories are built versatile due to different viewpoints. Classification can help to provide a more intuitive, individual-adopted visualization. For instance, categorization and clustering are used for building knowledge maps. A knowledge map is a visual representation of a set of knowledge objects and their relationships. Categorization can be used to populate knowledge maps with the (pointers to the) right documents supporting, e.g., a business process. Clustering enables the discovery of segments as knowledge objects of a map [BOE96].

Classification can not only be used to support visualization but can also ameliorate for example the results of document retrieval processes. One approach is the so called cluster-based retrieval which simplifies the retrieval process by using cluster representatives. This idea is based on the following hypothesis: Closely associated documents tend to be relevant for the same query [VRI79]. By using clustering techniques to ease retrieval, the clustering structure inherently provides an indexing scheme for retrieval and speeds up the retrieval time. The fundamental difference between cluster-based and non-cluster-based retrieval is illustrated in Figure 2.

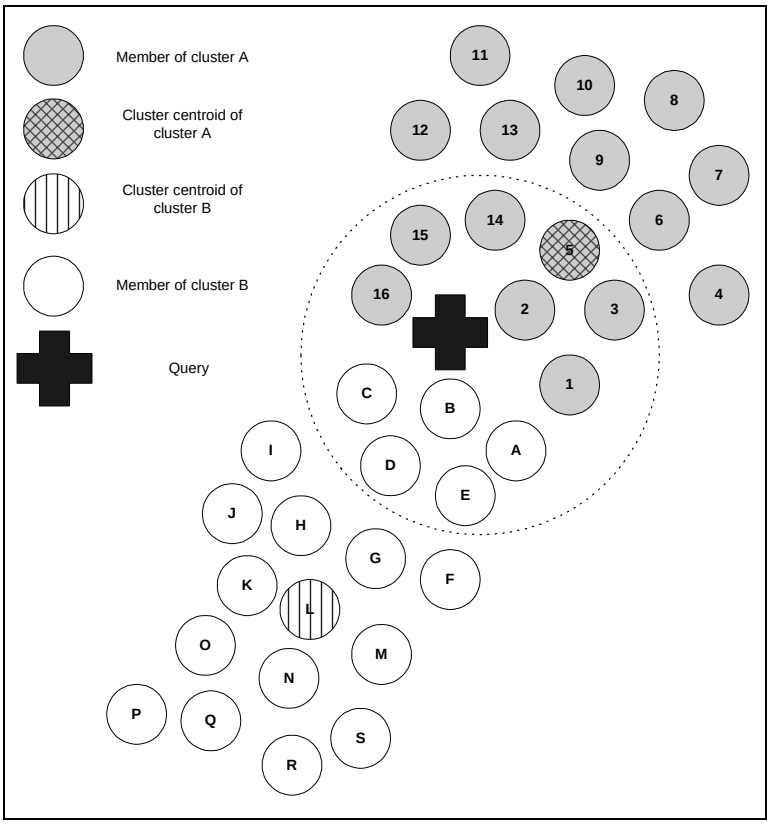

Figure 2: Example for Cluster-based Retrieval

Suppose a query is located at the boundary of two clusters, non cluster-based retrieval will include items from both clusters, i.e., items that are members of the dotted circle in Figure 2, at the top of the ranked list. The top ten of the ranked document list would appear as follows (we only list the documents index):

2, B, C, 16, 15, 14, 5, 3, 1, A

The cluster-based retrieval compares the distance between the query and each cluster centroid and ranks the items whose cluster centroid is nearer to the query at the top of the search result list. The ranked document list would be:

5, 3, 2, 14, 9, 6, 4, 1, 15, 13

Cluster-based retrieval methods based on hierarchical clustering are the most commonly used methods. One reason for this is that partitional clustering methods produce only useful results if the clustering is detailed. Document collections are typically large and, thus, a detailed clustering is often not desired. For this reason partitional strategies have not been aggressively pursued for improving information retrieval. Hierarchical clustering methods provide structures that are more versatile. They can deal with large document collections but, as already emphasized, have a shortcoming that could spoil the use of the clustering process: Their steps are not reversible [VRI79]. If the algorithm mistakenly merges two homogeneous clusters, the resulting cluster could be meaningless to the user. Furthermore, hierarchical cluster methods tend to produce elongated clusters partly due to the fact that they are not robust against outliers. Elongated clusters have the negative property that two objects can belong to the same cluster although their similarity is very small.

For document retrieval and organizing knowledge it may be useful to apply a clustering technique which results in non-exclusive cluster. Fuzzy clustering methods may be appropriate for this purpose [JAI99]. They have also the advantage of being able to handle mixed data types.

6 Summary and Outlook

In this paper we described the two classification approaches (i.e. categorization and clustering) and their precedent steps. For each approach we gave a brief description of the underlying theory and outlined the advantages and disadvantages of the different methods. We specified potential application areas in connection with knowledge management and illustrated exemplarily one topic in detail. We chose the enhancement of queries for illustration purposes because it is a common research problem with respect to information retrieval and, thus, to knowledge management. Table 3 recapitulates the subjects discussed in this paper.

Supervised learning (categorization) is far more important in text mining than in data mining. A main problem of categorization approaches is the time required for manually pre-categorizing the training data set by a human expert. Research has to be done with the focus of reducing the duration effort of the manual categorization phase. To date, there exist only a few ideas how this phase could be shortened. One proposition is to use the result of a clustering process as input for the catego-

rization training, i.e., to give the expert suggestions of potential categories and also names [YAN00].

Criteria	Document Categorization	Document Clustering
Document Preprocessing	Feature Extraction Feature Selection Document Representation	
Learning Type	Supervised, needs pre-categorized documents	Unsupervised, but needs number of clusters or similarity threshold
Important Methods	Decision Trees Decision Rules k-Nearest Neighbor Bayesian Approaches Neural Networks Regression-based methods Vector-based methods • Centroid • SVM	Hierarchical Methods • Single Link • Complete Link • Average Link • Ward's Method Partitional Methods • K-Means • Single-Pass • Nearest Neighbor • Expectation Maximization Fuzzy Clustering • Fuzzy C-Means
Knowledge Management Functions Supported	Retrieval Organization Visualization Development Exchange	

Table 3: Categorization vs. Clustering

Another approach is to shift the training duty of a human expert to the user. In a first step the categories are represented by a query. While using the retrieval system the user has to mark the retrieved documents as appropriate or not. This feedback is used to refine the category representatives [AUT01]. Further to reduce the users effort the class representative can automatically be improved by clustering former queries which covered the same topic. Those queries help to reformulate the class representative [BRU01].

7 References

[AND73] Anderberg, M. R. (1973): Cluster Analysis for Applications, Academic Press: New York.

[APT94] Apté, C., Damerau, F., Weiss, S. M. (1994): Towards Language Independent Automated Learning of Text Categorization Models, in: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 23-30.

[AUT01] Autonomy (2001): Technology White Paper, URL:http://www.autonomy.com/perl/Register.perl?doc=/autonomy_v3/Media/Collaterals/Autonomy_White_Papers/Autonomy_Technology_WP_0401.pdf [as of 2002-03-04].

[BOE96] Boersma, J., Stegwee, R. A. (1996): Exploring the issues in Knowledge Management, in: Proceedings of the 1996 IRMA International Conference, pp. 217-222.

[BOW01] Bowman, M. (2001): Text Processing, URL: http://www.cse.ogi.edu/class/cse580ir/handouts/23%20September/Text%20 Processing [2001-09-26].

[BRU01] Bruecher, H. (2001): Agentenbasiertes, dynamisches Benutzerportal im Wissensmanagement, Deutscher Universitätsverlag: Wiesbaden.

[DEM77] Dempster, A. P., Laird, M. N., Rubin, D. B. (1977): Maximum Likelihood from Incomplete Data via the EM algorithm, in: Journal of the Royal Statistical Society, Series B, Vol. 39, No. 1, pp. 1-38.

[DOE01] Doerre, J., Gerstl, P., Seiffert, R. (2001): Text Mining, in: Hippner, H., Kuesters, U., Meyer, M., Wilde, K. (Eds), Handbuch Data Mining im Marketing, Vieweg/Gabler: Braunschweig/Wiesbaden, pp. 465-488.

[DOM97] Domingos, P., Pazzani, M. (1997): On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, in: Machine Learning, Vol. 29, No. 2-3, pp. 103-130.

[DUB87] Dubes, R.C. (1987): How many clusters are best? - an experiment, in: Pattern Recognition, Vol. 20, No. 6, pp. 645–666.

[DUM98] Dumais, S. Platt, J., Heckermann, D., Sahami, M. (1998): Inductive Learning Algorithms and Representations for Text, in: Proceedings of the 7th International Conference on Information and Knowledge Management, pp. 148-155.

[FRA99] Frakes, W. B., Baeza-Yates, R. (1999): Modern Information Retrieval, Addison Wesley: New York.

[GER01] Gerstl, P., Hertweck, M., Kuhn, B. (2001): Text Mining: Grundlagen, Verfahren und Anwendungen, in: Praxis der Wirtschaftsinformatik- Business Intelligence, Vol. 39, No. 222, pp. 38-48.

[GOL85] Golden, B. L., Stewart, W. R. (1985): Empirical Analysis of Heuristics, in: Lawler, E. L., Lenstra, J. K., Rinooy Kan, A. H. G., Shmoys, D. B. (Eds), The Traveling Salesman Problem, Wiley: New York, pp. 207-249.

[GRE01] Greiner, R., Schaffer, J. (2001): AIxploratorium - Decision Trees, URL: http://www.cs.ualberta.ca/~aixplore/learning/DecisionTrees [2001-08-02].

[GUH98] Guha, S., Rastogi, R., Schim, K. (1998): CURE: An Efficient Clustering Algorithm for Large Databases, in: Proceedings of 1998 ACM-SIGMOD International Conference, Management of Data (SIGMOD'98), pp. 73-84.

[HEA98] Hearst, M. A., Schoelkopf, B., Dumais, S., Osuna, E., Platt, J. (1998): Trends and Controversies - Support Vector Machines, in: IEEE Intelligent Systems, Vol. 13, No. 4, pp. 18-28.

[HER01] Herrmann, K. (2001): Rakesh Agrawal: Athena: Mining-based Interactive Management of Text Databases, URL: http://www3.informatik.tu-muenchen.de/lehre/WS2001/HSEM-bayer/textmining.pdf [as of 2002-03-02].

[HOE97] Hoeppner, F., Klawonn, F., Kruse, R. (1997): Fuzzy Clusteringanalyse, Verfahren für die Bilderkennung, Klassifikation und Datenanalyse, Vieweg: Braunschweig et al.

[JAI88] Jain, A. K., Dubes, R. C. (1988): Algorithms for Clustering Data, Prentice Hall: Englewood Cliffs.

[JAI99] Jain, A. K., Murty, M. N., Flynn, P. J. (1999): Data Clustering: A Review, in: ACM Computing Surveys, Vol. 31, No. 3, pp. 264-323.

[JAR69] Jardine, N. (1969): Towards a General Theory of Clustering, in: Biometrics, Vol. 25, pp. 609-610.

[JOA98] Joachims, T. (1998): Text Categorization with Support Vector Machines: Learning with Many Relevant Features, in: Proceedings of the 10th European Conference on Machine Learning, pp. 137-142.

[JOH98] Johnson, R. A., Wichern, D. W. (1998): Applied Multivariate Statistical Analysis, Prentice Hall: Englewood Cliffs.

[KRA98] Krahl, D., Windheuser, U., Zick, F.-K. (1998): Data Mining - Einsatz in der Praxis, Addison Wesley Longman: Bonn.

[KRI94] Krishnapuram, R., Keller, J. M. (1994): Fuzzy and Possibilistic Clustering Methods for Computer Vision, in: Mitra, S., Gupta, M., Kraske, W. (Eds), Neural and Fuzzy Systems, SPIE Institute, pp. 133-159.

[LAM97] Lam, W., Low, K. F., Ho, C. Y. (1997): Using a Bayesian Network Induction Approach for Text Categorization, in: Proceedings of the 15th International Joint Conference on Artificial Intelligence, pp. 745-750.

[LAM98] Lam, W., Ho, C. Y. (1998): Using a Generalized Instance Set for Automatic Text Categorization, in: Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 81-89.

[LAR96] Larkey, L. S., Croft, W. B. (1996): Combining Classifiers in Text Categorization, in: Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 289-297.

[LEW94] Lewis, D. D., Ringuette, M. (1994): A Comparison of Two Learning Algorithms for Text Categorization, in: Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, pp 81-93.

[MAD01] Madani, O. (2001): ABCs of Text Categorization,
URL: http://classes.seattleu.edu/computer_science/csse470/Madani/ABCs.html
[2001-04-24].

[MIR96] Mirkin, B. (1996): Mathematical Classification and Clustering, Kluwer Academic Publishers: Dordrecht et al.

[MYA01] Myax Knowledge Management (2001): Knowledge Management im Unternehmen I: Erfolgreiche Wissensakquisition im Internet, Paper presented at a Workshop in Zürich.

[NGG97] Ng, H. T., Goh, W. B., Low, K. L. (1997): Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization, in: Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 67-73.

[NIE95] Nie, J. H. (1995): Constructing Fuzzy Model by Self-Organizing Counterpropagation Network, in: IEEE Transactions on Systems, Man and Cybernetics, Volume 25, No. 6, pp. 963-970.

[RUI98] Ruiz, M. E., Srinivasan, P. (1998): Automatic Text Categorization Using Neural Network, in: Proceedings of the 8th ASIS SIG/CR Workshop on Classification Research, pp. 59-72.

[SHE79] Shepard, R. N., Arabie, P. (1979): Additive Clustering: Representation of Similarities as Combinations of Discrete Overlapping Properties, in: Psychological Review, Vol. 86, pp. 87-123.

[SIO00] Siolas, G., d'Alché-Buc, F. (2000): Support Vector Machines based on a semantic kernel for text categorization, in: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00), pp. 205-209.

[VRI79] van Rijsbergen, C. J. (1979): Information Retrieval, 2nd ed., URL: http://www.dcs.gla.ac.uk/Keith/Preface.html#PREFACE [as of 2001-12-21].

[VRY77] van Ryzin, J. (1977): Classification and Clustering, Academic Press: New York.

[WAR63] Ward, J. H. (1963): Hierarchical grouping to optimize an objective function, in: Journal American Statistic Association, Vol. 58, pp. 236-244.

[WEI01] Wei, C. P, Dong, Y. X. (2001): A Mining-Based Category Evolution Approach to Managing Online Document Categories, in: Proceedings of the 34th Annual Hawaii International Conference on System Sciences.

[WIT00] Witten, I. H., Frank, E. (2000): Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers: San Francisco.

[YAN94] Yang, Y., Chute, C. (1994): An Example-Based Mapping Method for Text Categorization and Retrieval, in: ACM Transactions on Information Systems, Vol. 12, No. 3, pp. 253-277.

[YAN99] Yang, Y., Liu, X. (1999): A Re-Examination of Text Categorization Methods, in: Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 42-49.

[YAN00] Yang, H.-C., Lee, C.-H. (2000): Automatic Category Generation for Text Documents by Self-Organizing Maps, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00), pp. 581-586.

[ZAD65] Zadeh, L. A. (1965): Fuzzy sets, in: Information Control, Vol. 8, No. 3, pp. 338-353.

[ZAH71] Zahn, C. T. (1971): Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters, in: IEEE Transactions on Computers, Vol. 20, No. 1, pp. 68-86.

[ZHO01] Zhou, W., Maerz, N. H. (2001): Multivariate Clustering Analysis of Discontinuity Data: Implementation and Applications, in: Proceedings of the 38th U.S. Rock Mechanics Symposium, pp. 861-868.