

Restrictive Clustering and Metaclustering for Self-Organizing Document Collections

Stefan Siersdorfer, Sergej Sizov
{stes, sizov}@mpi-sb.mpg.de
Max Planck Institut fuer Informatik
66123 Saarbruecken, Germany

ABSTRACT

This paper addresses the problem of automatically structuring heterogeneous document collections by using clustering methods. In contrast to traditional clustering, we study restrictive methods and ensemble-based meta methods that may decide to leave out some documents rather than assigning them to inappropriate clusters with low confidence. These techniques result in higher cluster purity, better overall accuracy, and make unsupervised self-organization more robust. Our comprehensive experimental studies on three different real-world data collections demonstrate these benefits. The proposed methods seem particularly suitable for automatically substructuring personal email folders or personal Web directories that are populated by focused crawlers, and they can be combined with supervised classification techniques.

Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval—*clustering*

General Terms

Algorithms, Theory

Keywords

Meta Clustering, Restrictive Clustering

1. INTRODUCTION

1.1 Problem

This paper addresses the problem of automatically structuring heterogeneous document collections into thematically coherent subsets. This issue is relevant for a variety of applications, such as organizing large personal email folders, dividing topics in large Web directories into subtopics, structuring large amounts of company and intranet data,

etc. The methods of choice for accomplishing this are either based on supervised classification, which requires explicit, manually labeled, training data, or unsupervised clustering. In many situations explicit training data is unavailable, so that clustering is the only viable option.

Conventional clustering methods partition the entire data set into clusters, but this may lead to poor results in terms of cluster impurity, for example, mixing thematically unrelated documents into the same cluster. The approach that we advocate and further develop in this paper is to cluster only a *subset* of the available data, but do so with a higher clustering quality. The left-out data which does not assigned to any cluster is collapsed into an extra container with "miscellaneous" documents. We call this approach *restrictive clustering* methods. With *simple restrictive methods* we provide restrictive modifications of conventional clustering methods; with *meta-clustering methods* we combine different clustering methods in a restrictive way. Additionally, by using supervised learning techniques (e.g., SVM-based text classification) on the results of the restrictive clustering methods and meta methods, we can generalize the clusters to a larger subset or even the entire dataset.

As a possible application scenario for our techniques consider a focused Web crawler [19]. Such a crawler starts with a set of training documents for a given topic or an entire topic directory, e.g., for topic "sports" with subtopics "ball games", "track and fields", and "swimming". The result of a large crawl populate these explicitly labeled classes, and we may obtain a huge number of documents in the ball games topics and a much smaller number of documents in the other two subclasses. Obviously this suggests that the originally given topic directory was not wisely chosen and should have foreseen additional subsubtopics under the ball games class. What we would like to achieve now is an automatic, but unsupervised, organization of the ball games documents by partitioning this class into appropriate subsubclasses. In doing this we strive for high accuracy in the sense that whatever subsubclasses we form should indeed be reasonably homogeneous, but it would be perfectly acceptable to completely leave out documents for which a cluster assignment can be made only with very low confidence. These left-out documents would simply be considered as a subsubtopic "/sports/ball games/miscellaneous". Suppose we can create three new clusters that correspond to "soccer", "basketball", and "handball" documents. Initially, these clusters would be unlabeled, but if each of them has very high thematic purity then the user could easily assign labels after inspecting a few samples from each class (or additional methods based

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.,
Copyright 2004 ACM 1-58113-881-4/04/0007 ...\$5.00.

on term statistics analysis could automatically suggest appropriate class labels). This is when the restrictive nature of our clustering methods and the resulting higher class purity, relative to traditional clustering, would pay off towards making personal data collections self-organizing.

1.2 Related Work

Combining multiple clustering methods in an ensemble learning manner has been addressed in [20, 9]. Neither of these papers considers restrictive methods where documents may be completely left out and are not assigned to any cluster; we believe that this is crucial for aiming at very high precision. Also, none of the prior work provides analytical models, which is crucial for understanding why such methods work. Finally, our application context is broader and combines meta clustering with other techniques like supervised classification, and we present much more comprehensive application-oriented experimental results with real-life datasets.

1.3 Contribution and Outline

Our contribution consists of five points. First, we develop a systematic and comprehensive framework for restrictive clustering methods and meta methods. Second, we identify a particularly beneficial technique, coined *metamapping*, where we combine the clusters found under different simple clustering methods. Third, we provide a simple probabilistic model that explains why the meta technique improves accuracy at the expense of “losing” some fraction of documents (which will then be organized into the “miscellaneous” container); the model could even be used for approximately predicting the achievable accuracy and loss of our techniques. Fourth, we show how to combine the restrictive clustering methods and meta methods with standard classification such as SVM. Fifth, we provide a comprehensive experimental study of the pros and cons of a variety of methods, including our *metamapping* technique and also transductive SVMs.

The rest of the paper is organized as follows. In Section 2 we briefly review the technical basics of clustering methods. Section 3 presents our notion of restrictive methods: we describe simple restrictive methods and the restrictive combination of different clustering methods. In Section 4 we combine restrictive clustering and supervised learning. Section 5 provides experiments on different real-world datasets.

2. TECHNICAL BASICS

Clustering algorithms partition a set of objects, text documents in our case, into groups called *clusters*. For clustering we represent documents as feature vectors. In the prevalent bag-of-words model the features are derived from word occurrence frequencies [4, 15] (e.g. capturing tf or $tf * idf$ weights of terms). In addition, feature selection algorithms [14] can be applied to reduce the dimensionality of the feature space and eliminate “noisy”, non-characteristic features, based on term frequencies or advanced information-theoretic measures for feature ordering (e.g., mutual information (MI) or information gain [14]).

Clustering methods

Clustering methods can be divided into the following groups [8]: partitioning methods, hierarchical methods, density based methods, grid based methods, model based methods. In

this paper we consider partitioning methods: the dataset is divided into disjoint partitions. The number k of clusters is a tuning parameter for this family of clustering algorithms [10].

A simple, very popular member of the family of partitioning clustering methods is *k-Means* [11]: k initial centers (points) are chosen, every document vector is assigned to the nearest center (according to some distance or similarity metric), and new centers are obtained by computing the means (centroids) of the sets of vectors in each cluster. After some iterations (according to a stopping criterion) one obtains the final centers, and one can cluster the documents accordingly. A similar algorithm, which can be considered as a “smoothed” form of *k-Means* is *EM clustering* [10, 15]: in every iteration the probabilities of the objects for being contained in the different clusters are updated using the expectation-maximization technique.

The result and run-time for *k-Means* and other iterative clustering algorithms are strongly dependent on the initial partitioning (for *k-Means* this corresponds to the initial centers). A standard heuristics for this initialization phase is *preclustering* [8]: before starting the actual clustering algorithm, a clustering is computed on a much smaller subset. This way one can often obtain better starting points.

The method of *singular value decomposition* (SVD) [5] on a document corpus transforms the initial document-term space into a lower dimensional “topic”-term space. One of the results of SVD is a document-“topic” similarity matrix; so the naive way to perform a clustering would be to assign every document to one of the top- k SVD “topics” (corresponding to the k largest singular values). [12] describes an SVD-based method which results in better clustering quality: by using the SVD of the term-document-matrix, one can transform the document-vectors in a new feature space and then apply *k-Means* on the transformed vectors.

Feature selection

Mutual Information (MI) has been shown to be an effective method for feature selection [14]. MI, when applied to a document corpus, requires an a priori partitioning of documents into thematically coherent subsets. If this partitioning is not already given, a clustering algorithm can provide us with an initial approximation. This initial step may use either all features or a df (document frequency) based feature selection. Once we have clusters, we can compute MI values and identify the most discriminative features, and then we can iterate this procedure, alternating between feature selection and clustering. We developed an iterative clustering algorithm based on this approach, using *k-Means* as the underlying base method.

Supervised learning methods

In contrast to unsupervised clustering methods discussed above, supervised learning methods use labeled training documents to build a classification model. For instance, linear support vector machines (SVMs) [7] construct a hyperplane $\vec{w} \cdot \vec{x} + b = 0$ that separates the set of positive training examples from a set of negative examples with maximum margin. For a new, previously unseen, document \vec{d} the SVM merely needs to test whether the document lies on the “positive” side or the “negative” side of the separating hyperplane. Transductive support vector machines (TSVM) [21] take, in addition to the training documents, unlabeled documents

into account. In Section 4 we will present a method to combine such supervised or semisupervised learning methods with introduced *restrictive* clustering algorithms.

3. RESTRICTIVE CLUSTERING

3.1 Making Simple Methods restrictive

The idea of restrictive clustering is to avoid making a decision about a document at all if that decision can be made only with low confidence. So out of a given set of unlabeled data U , our method chooses a subset S of documents that are assigned to clusters, and abstains on the documents in $U - S$. We call the ratio $|U - S|/|U|$ of dismissed documents the document *loss*.

We can use confidence measures to make simple methods restrictive. For the different variants of the k-Means method a natural confidence measure is the distance of a document vector from the nearest centroid (or some other similarity measure). So we can tune these methods by requiring that accepted or rejected documents have a distance above some threshold, and abstain otherwise. The threshold is our tuning parameter.

Given an application-acceptable loss of L percent, we can make a clustering method restrictive by dismissing the L percent of the documents with the lowest confidence values. Although this is a fairly straightforward idea, we are not aware of prior literature that has explicitly considered such restrictive clustering methods.

3.2 Restrictive Meta Methods

For meta clustering we are given a set $C = \{c_1, \dots, c_l\}$ of different clustering methods. A document d is assigned to one of k clusters with labels $\{1, \dots, k\}$: $c_i(d) \in \{1, \dots, k\}$. The idea of meta clustering is now to combine the different clustering results in an appropriate way.

3.2.1 Metamapping

To combine the $c_i(d)$ into a metaresult, the first problem is to determine which cluster labels of different methods c_i correspond to each other. (Note that cluster label 2 of method c_i does not necessarily correspond to the same cluster label 2 of method c_j , but could correspond to say cluster label 5.) With perfect clustering methods the solution would be trivial: the documents labeled by c_i as a would be exactly the documents labeled by c_j as b , and we could easily test this with one representative per cluster. This assumption is, of course, unrealistic; rather clustering results exhibit a certain fuzziness so that some documents end up in clusters other than their perfectly suitable cluster. Informally, for different clustering methods we would like to associate the clusters which each other which are “most correlated”¹.

Formally, for every method c_i we want to determine a bijective function $map_i : \{1, \dots, k\} \rightarrow \{1, \dots, k\}$ which assigns all labels $a \in \{1, \dots, k\}$ assigned by c_i a meta label

¹One possible way to obtain the final clustering could be to consider sets of labels assigned to each document by particular algorithms as new document-specific feature vectors and apply final clustering (say, using k-Means) in this new feature space. However, the components of such feature vectors are in fact labels and not numbers. It is not clear how to define proper distance/similarity measures between such vectors for clustering and - optionally - for restrictive removal of ‘weak’ candidates as proposed in Section 3.1.

$map_i(a)$. By these mappings the clustering labels of the different methods are associated with each other and we can define the clustering result for document d using method c_i as:

$$result_i(d) := map_i(c_i(d)) \quad (1)$$

We now describe different ways to obtain the map_i functions.

Method A: Correlation-based Approach

We want to maximize the correlation between the cluster labels. For sets $A_1..A_x$, we can define their *overlap* as

$$overlap(A_1, \dots, A_x) := \frac{|A_1 \cap \dots \cap A_x|}{|A_1| + \dots + |A_x| - |A_1 \cap \dots \cap A_x|} \quad (2)$$

Now using

$$A_{ij} := \{d \in U | res_i(d) = j\} \quad (3)$$

we can define the *average overlap* for a document set U and the set of clustering methods C as

$$\frac{1}{k} \sum_{j=1}^k \frac{1}{\binom{l}{j}} \sum_{(i,m) \in \{1, \dots, k\}^2, i < m} overlap(A_{ij}, A_{mj}) \quad (4)$$

We are interested in the mappings map_i which maximize the average overlap. However there is a combinatoric explosion: there are $k!^{l-1}$ possibilities to build mappings. A greedy approach is to maximize the overlap between pairs of clustering methods, e.g. c_1 and c_2 , c_2 and c_3 , ..., c_{k-1} and c_k , and to use transitivity to compute an overall mapping. An even greedier approach is to find for all c_{i-1} and c_i the highest, second highest, third highest, etc. $overlap(A_{i-1,j}, A_{ij})$, to derive the mapping for c_{i-1} and c_i and to compute the overall mapping using again transitivity.

Method B: Purity-based Approach

Instead of maximizing average overlap for all mappings, one may consider a ranked list of meta clusters, ordered by their “purity”. The underlying idea of this approach is to prioritize the clusters that produce a high overlap only in one particular (potentially proper) combination and low overlaps otherwise. The functions map_i for all methods c_i are constructed for particular meta labels $\{1..k\}$ in an iterative manner (i.e., we identify in each step all particular clusters to be associated with the given meta label $m = 1, \dots, k$).

Starting with $m = 1$, we construct for every cluster A_{ij} the set of all possible mapping functions that would map this cluster (and some clusters from other methods) onto label m . Each combination corresponds to a set of partial cluster candidates for this label:

$$Q_x^m(A_{ij}) = \{A_{1j_1}, A_{2j_2}..A_{ij}..A_{lj_l}\}, jk \in 1..l, j_k \neq j \quad (5)$$

where A_{ij} is fixed and all other elements are variable for different x . The normalized variance

$$purity(A_{ij}) = \frac{var(overlap(Q_x^m(A_{ij})))}{max\{overlap(Q_x^m(A_{ij}))\}} \quad (6)$$

characterizes the purity of A_{ij} : this value is higher for A_{ij} with higher specificity and lower for poor (in the worst case, randomly generated) classes. The normalization is used to make purity values comparable for different A_{ij} .

It is natural to consider the cluster A_{ij}^* with maximum value of $purity(A_{ij}^*)$ as a “good” cluster. Now we fix the

association of A_{ij}^* with label m and continue the purity-based comparison to identify the second, third, and further remaining clusters from other methods that should be associated with the same label. To reduce the computational overhead, the whole collection $Q_x^m(A_{ij}^*)$ with highest overlap for A_{ij}^* can be associated with label m just after the first “good” cluster A_{ij}^* is identified.

We notice that clusters from particular methods are selected in the order of their purity values. Thus, this approach can be also used to produce the natural ranking of clustering methods. This option can be useful to recognize when clustering methods fail (for instance, their results could be excluded from meta mappings).

When the cluster mapping for the given label m is complete, all selected clusters are associated with label m and excluded from further consideration. Now, the same procedure can be repeated for label $m + 1$ on remaining clusters and continued until full bijections $map_i : \{1..k\} \rightarrow \{1..k\}, i = 1..l$ are complete.

Method C: Association-rules-based Approach

Another approach is to use association rules mining [3, 10], which was popularized for market basket analysis. Our (shopping) items here would be tuples $(c_i, c_i(d))$ of clustering methods and cluster labels. To every document $d \in U$ we can now assign an itemset:

$$\{(c_1, c_1(d)), \dots, (c_k, c_k(d))\} \quad (7)$$

For these itemsets we can now apply a data mining algorithm like the well known Apriori algorithm [3] to compute a ranked list of association rules. As an example we could obtain the following rule:

$$\{(c_5, 3)\} \implies \{(c_2, 4), (c_3, 3)\} \quad (8)$$

This can be interpreted as: “Cluster label 3 of method c_5 corresponds to cluster label 4 of c_2 and cluster label 3 of c_3 .”

Now starting with the highest ranked rule we can process the list of rules by successively deducing our mappings this way until we have obtained all mappings (ignoring all mapping proposals from lower ranked rules which contradict previous, higher ranked mappings).

As in method A we can apply here also a greedier approach by computing the mapping for pairs of clustering methods and using transitivity to find the overall mapping.

The introduced approaches were designed for methods with constant pre-defined number of resulting clusters (e.g. density-based clustering algorithms, such as k-Means). The generalization for methods with a variable number of clusters (e.g., tree-based clustering approaches) is subject of our current work.

3.2.2 Metafunctions

After having computed the mapping we are given a set $C = \{C_1, \dots, C_l\}$ of l binary clustering methods with results $res_i(d)$. For simplicity we consider here the case of $k = 2$ clusters and choose $res_i(d) \in \{+1, -1\}$ for a document d , namely, +1 if d is assigned to cluster 1, and -1 if d is assigned to cluster 2. We can combine these results into a meta result: $Meta(d) = Meta(res_1(d), \dots, res_l(d))$ in $\{+1, -1, 0\}$ where 0 means abstention. A family of such meta methods is the linear combination with thresholding [17]. Given thresholds t_1 and t_2 , with $t_1 > t_2$, and weights $w(c_i)$ for the l underlying

clustering methods we compute $Meta(d)$ as follows:

$$Meta(d) = \begin{cases} +1 & \text{if } \sum_{i=1}^l res_i(d) \cdot w(c_i) > t_1 \\ -1 & \text{if } \sum_{i=1}^l res_i(d) \cdot w(c_i) < t_2 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

This meta clustering family has some important special cases, depending on the choice of the weights and thresholds:

- 1) voting (analog to bagging [6] in supervised learning): Meta returns the result of the majority of the clustering methods.
- 2) unanimous decision: if all methods yield the same result (either +1 or -1), Meta returns this result, 0 otherwise.
- 3) weighted averaging: Meta weighs the clustering methods by using some predetermined quality estimator.

These considerations can be easily generalized to the case of $k > 2$ possible clusters.

The restrictive and tunable behavior is achieved by the choice of the thresholds: we dismiss the documents where the linear result combination lies between t_1 and t_2 . In the rest of the paper we will consider only the unanimous-decision meta method as the simplest of the above cases in order to demonstrate the feasibility of our approach. The approach itself carries over to more sophisticated instantiations of the meta framework.

3.2.3 A probabilistic model for metaclustering

In this subsection we develop a simplified probabilistic model (for $k = 2$) to a better understanding of why meta-clustering works. Consider the unanimous-decision clustering meta method defined above. We assume that we have found appropriate mappings map_i as described above. We associate a Bernoulli random variable X_i with each clustering method c_i , where $X_i = 1$ if c_i clusters a document correctly, 0 otherwise.

From basic probability theory it follows that

$$P(X_1 = 1 \wedge X_2 = 1) = cov(X_1, X_2) + P(X_1 = 1) * P(X_2 = 1) \quad (10)$$

where

$$cov(X_1, X_2) = \frac{1}{n-1} \cdot \sum_j (x_1 - \bar{x}_1)(x_2 - \bar{x}_2) \quad (11)$$

is the *covariance* for the data points (x_1, x_2) of the joint distribution of (X_1, X_2) .

To model the case of $l > 2$ clustering methods we use a tree dependence model, which is a well known approximation method in probabilistic IR ([16]). We define a *Dependence Graph* $G = (V, E)$ where V consists of the Bernoulli variables X_i and which contains for all X_i, X_j ($i \neq j$) an undirected edge $e(X_i, X_j)$ with weight $w(e(X_i, X_j)) = cov(X_i, X_j)$. We approximate the Dependence Graph by a maximum spanning tree $G' = (V, E')$ which maximizes the sum of the edge weights. The nodes in G' with no edges in between are considered as independent. So we obtain:

$$P(X_1 = x_1, \dots, X_k = x_l) = P(X_{root} = 1) \prod_{(i,j) \in E'} \frac{P(X_i = x_i, X_j = x_j)}{P(X_i = x_j)} \quad (12)$$

where X_{root} is the root node of the tree G' and $x_i \in \{0, 1\}$.

Now we introduce the following special case: For any two clustering methods c_i, c_j the covariance has approximately

the same value cov . With $w(e(X_i, X_j)) = cov$ we can (without loss of generality) choose X_1 as the root node and the edges (X_i, X_{i+1}) as tree edges.

Now we have:

$$\begin{aligned} P(X_1 = 1, \dots, X_l = 1) &= P(X_1 = 1) \prod_{i=1}^{l-1} P(X_{i+1} | X_i) \\ &= P(X_1 = 1) \prod_{i=1}^{l-1} \frac{P(X_i = 1, X_{i+1} = 1)}{P(X_i = 1)} \end{aligned} \quad (13)$$

By considering equation 10 and Assumption 1 we obtain:

$$\begin{aligned} P(X_1 = 1, \dots, X_l = 1) &= \\ P(X_1 = 1) \prod_{i=1}^{l-1} \frac{P(X_i = 1)P(X_{i+1} = 1) + cov}{P(X_i = 1)} \end{aligned} \quad (14)$$

Analogously we obtain $P(X_1 = 0 \wedge \dots \wedge X_l = 0)$.

Finally we can substitute these results in the following formulas for the loss probability

$$\begin{aligned} loss(Meta) &= 1 - P(X_1 = \dots = X_l) \\ &= 1 - (P(X_1 = 1, \dots, X_l = 1) + P(X_1 = 0, \dots, X_l = 0)) \end{aligned} \quad (15)$$

and the error and accuracy probability

$$\begin{aligned} error(Meta) &= P(X_1 = 0 \dots X_l = 0 | X_1 = \dots = X_l) \\ &= \frac{P(X_1 = 0 \dots X_l = 0)}{P(X_1 = 1 \dots X_l = 1) + P(X_1 = 0 \dots X_l = 0)} \end{aligned} \quad (16)$$

$$accuracy(Meta) = 1 - error(Meta) \quad (17)$$

As an illustrative example we consider the case that the l clustering methods have the same probability $p < 0.5$ (i.e. the clustering method performs better than random) to misassign a document, and a covariance $c < 1.0$ (i.e. the clustering methods are not perfectly correlated). In this case we would obtain for $loss$ and $error$:

$$loss = 1 - \left((1-p) \left(\frac{c + (1-p)^2}{1-p} \right)^{l-1} + p \left(\frac{c+p^2}{p} \right)^{l-1} \right) \quad (18)$$

$$error = \frac{p \left(\frac{c+p^2}{p} \right)^{l-1}}{(1-p) \left(\frac{c+(1-p)^2}{1-p} \right)^{l-1} + p \left(\frac{c+p^2}{p} \right)^{l-1}} \quad (19)$$

It is easy to show that for $l \rightarrow \infty$ the loss converges monotonically to 1, and the error to 0 (i.e. with more clustering methods we can obtain a lower error but pay the price of a higher loss). The covariance plays the role of a “smoothing constant”: with higher correlated clustering methods the convergence of both loss and error is slowed down.

4. COMBINATION OF RESTRICTIVE CLUSTERING WITH SUPERVISED LEARNING

The partitioning produced by a restrictive meta algorithm is expected to have a higher accuracy than the results of the underlying (non-restrictive) base methods. However, the higher clustering quality is connected with the loss of

more or less data. This situation is acceptable in precision-oriented applications (e.g. data filtering) but may cause problems in recall-sensitive cases. To overcome this limitation of the restrictive clustering, the filtered output of the meta algorithm can be considered as training input for supervised or semisupervised classification methods (e.g. Naive Bayes, SVM, transductive SVM etc.). The latter method allows customizable partitioning of unlabeled data (usually proportional to sizes of labeled data sets).

The collection of documents that were rejected by meta clustering can be assigned to clusters using this new decision model. It is obvious that the combination of restrictive meta-clustering and classification acts as a non-restrictive clustering approach (with zero loss). In Chapter 5, we show results of preliminary evaluations for meta clustering in connection with linear SVM and transductive SVM algorithms.

5. EXPERIMENTS

5.1 Quality Metrics for Clustering

Our quality measure describes the correlation between the actual topics of our datasets and the clusters found by the algorithm. Consider that the clusterlabels can be permuted: Given two classes $class_1$ and $class_2$, it does not matter for example whether a clustering algorithm assigns label a to all documents contained to $class_1$ and label b contained in $class_2$ or vice versa; the documents belonging together are correctly put together and the quality should reach its maximum value (i.e. 1) and the error should be 0.

Let k be the number of classes and clusters, N_i the total number of clustered documents in $class_i$, N_{ij} the number of documents contained in $class_i$ and having clusterlabel j . We define:

$$accuracy = \max_{(j_1, \dots, j_k) \in perm((1, \dots, k))} \frac{\sum_{i=1}^k N_{i, j_i}}{\sum_{i=1}^k N_i} \quad (20)$$

and

$$error = 1 - accuracy \quad (21)$$

As $loss$ we simply define the fraction of documents dismissed over the whole document set. We use the macro-average of loss and error as an aggregation measure for a larger number of experiments.

5.2 Setup

We performed a series of experiments with real-life data from

1) Newsgroups collection at [1]. This collection contains 18828 articles collected from 20 Usenet newsgroups. Particular topics ('rec.autos', 'sci.space', etc.) contain between 600 and 1000 documents.

2) Reuters articles [13]. This is the most widely used test collection for text categorization research. The collection contains 21578 Reuters newswire stories, subdivided into multiple categories (e.g. 'earn', 'grain', 'trade').

3) Internet Movie Database (imdb) at [2]. Documents of this collection are short movie descriptions that include the storyboard, cast overview, and user comments. This collection contains 20 topics according to particular movie genres (e.g. 'drama', 'horror').

In all discussed experiments, the standard bag-of-words model [4] (using term frequencies to build L1-normalized feature vectors) was used for document representation.

Our experiments capture the behavior of (restrictive) base clustering methods and meta clustering, for tuples of topics such as "Drama vs. Horror vs. Western" for imdb data or "rec.autos vs. rec.motorcycles vs. rec.sport.hockey" for the newsgroups data. For each data set we identified all topics with sufficiently many documents. These were 20 topics for newsgroups, 15 for reuters and 15 for the genres of imdb documents. We randomly chose 50 topic pairs from newsgroups, from imdb, and from reuters for every set of k -tuples ($k = 2, 3, 5$). Finally, we computed macro-averaged results for these topic tuples.

5.3 Results

In our Experiments we considered the following base methods (see Section 2):

1. *base1*: k-Means, no feature selection, preclustering with $k * 20$ documents
2. *base2*: iterative feature selection applied on k-Means, preclustering with $k * 20$ documents on a preselected feature space (df), after each iteration: feature selection (step 1: top-2000 according to df , step 2: top-500 according to MI), number of iterations: 5
3. *base3*: transforming feature vectors using SVD (SVD rank = 2), application of k-Means on the transformed vectors (We found that a higher SVD rank results in a lower clustering accuracy in consistence with observations made by [12].)

Of course, the introduced meta approach can be used with any other clustering methods as well.

Figure 2 shows the loss-error tradeoff for the base methods for $k = 3$ and $k = 5$: By inducing a loss as described in Section 3.1 we can obtain a significant reduction of the error.

With the three base methods we built a restrictive meta classifier based on the "unanimous decision" function (Section 3.2.2) and the 3 different meta-mappings described in Section 3.2.1 namely:

1. *MapA*: correlation-based mapping
2. *MapB*: purity-based mapping
3. *MapC*: mapping using association rules

We compared the meta results with the results of the underlying base methods and the restrictive base methods (inducing the same loss as the meta method in each experiment). The results are shown in Figure 1. The results clearly show that the meta approach provides a lower error than its underlying base methods at the cost of moderate loss. More important, the meta method performs typically better than the restrictive version of each base method for the same loss. Figure 1 also shows that mappings produced by particular meta algorithms are not always identical; this leads to different losses on the same dataset.

Although all proposed meta-mapping algorithms perform fairly well and predict the same mapping in most of the cases, there are marginal differences for experiments on topics that are very different in size (especially reuters). In such cases the overlap mapping (MapA) tends to produce slightly better results. Otherwise, the purity-based mapping (MapB) is more stable in experiments with fairly comparable class sizes (newsgroups and imdb). The underlying Apriori algorithm of the association rule-based mapping (MapC)

has calibration parameters that, depending on properties of the current data, may lead to slightly less accurate results. The optimal, task-dependent, parameterization of this algorithm is subject of our future work.

To test the combination of clustering and supervised or semisupervised learning we performed a restrictive meta-clustering for $k = 2$ (using *MapB*) as described in Section 4. The obtained new partitioning for the whole document set (i.e. loss = 0) was compared to the clusterings provided by the underlying non-restrictive base methods. The evaluation is shown in Figure 3. Although the partitioning provided by restrictive meta-clustering has high accuracy and results in many cases in good training sets and accurate classifiers, the high loss (and small training sets) causes, in particular experiments, reduced generalization performance leading to moderate average accuracy.

6. CONCLUSIONS AND FUTURE WORK

In this paper we proposed an approach for automatically structuring heterogenous document collections by using restrictive clustering methods. A key element in our approach has been to construct restrictive meta methods that result in higher cluster purity. The introduced algorithms of meta mapping ensure better accuracy and make clustering results robust and accurate at the cost of moderate loss of uncertain samples. While the introduced approach applies to a wide range of partitioning methods, we have especially worked on k-Means and its advanced modifications. We have experimentally shown that meta clustering has higher accuracy than particular clustering methods and, more important, performs better than the restrictive version of each underlying base method with the same loss.

Our ongoing and future work includes

1. the generalization towards weighted meta methods (instead of simple unanimous strategy in the current approach) that may lead in lower loss with same accuracy,
2. recognition of failed clustering attempts and exclusion of potentially incorrect results from meta mapping and
3. application studies in the context of focused crawling and self-organized personal ontologies. The work presented here is embedded in the BINGO! project [18], a tool suite for building information portals and specialized search engines.

Our long-term goal is to develop robust and accurate unsupervised algorithms for information management and expert Web search.

7. REFERENCES

- [1] The 20 newsgroups data set. <http://www.ai.mit.edu/~jrennie/20Newsgroups/>.
- [2] Internet movie database. <http://www.imdb.com>.
- [3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994.
- [4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
- [5] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. Technical Report UT-CS-94-270, 1994.
- [6] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [7] C. Burges. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.
- [8] M. Ester, H.-P. Kriegel, and J. Sander. *Knowledge Discovery in Databases*. Springer, 2001.
- [9] A. Fred and A. K. Jain. Robust data clustering. In *Proc. Conference on Computer Vision and Pattern Recognition, CVPR*, 2003.
- [10] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.
- [11] J. Hartigan and M. Wong. A k-Means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.
- [12] M. Hasan and Y. Masumoto. Document clustering: before and after the singular value decomposition. Technical Report Information Processing Society of Japan, Natural Language, No.134, 1999.
- [13] D. D. Lewis. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318. Defense Advanced Research Projects Agency, Morgan Kaufmann, Feb. 1991.
- [14] W. Madison, Y. Yang, and J. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, 1997.
- [15] C. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
- [16] C. V. Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33:2, pp. 106–119, 1977.
- [17] S. Siersdorfer and S. Sizov. Construction of feature spaces and meta methods for classification of Web documents. *Conference on Database Systems for Business, Technology and Web (BTW)*, 2003.
- [18] S. Sizov, M. Biwer, J. Graupmann, S. Siersdorfer, M. Theobald, G. Weikum, and P. Zimmer. The BINGO! system for information portal generation and expert Web search. *Conference on Innovative Systems Research (CIDR)*, 2003.
- [19] S. Sizov, S. Siersdorfer, M. Theobald, and G. Weikum. BINGO!: Bookmark-induced gathering of information. *International Conference on Web Information Systems Engineering (WISE)*, 2002.
- [20] A. Strehl and J. Gosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3, pp. 583–617, 2002.
- [21] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

| k = 3 | | | | | | | | | |
|-------|-----------|----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|---------|
| Map | Meta | | restrictive Base | | | Base | | | Dataset |
| | avg(loss) | avg(err) | base1 avg(err) | base2 avg(err) | base3 avg(err) | base1 avg(err) | base2 avg(err) | base3 avg(err) | |
| MapA | 0,497 | 0,234 | 0,275 | 0,277 | 0,250 | 0,339 | 0,312 | 0,337 | IMDB |
| MapB | 0,542 | 0,229 | 0,276 | 0,274 | 0,232 | | | | |
| MapC | 0,458 | 0,236 | 0,287 | 0,281 | 0,263 | | | | |
| MapA | 0,420 | 0,242 | 0,269 | 0,304 | 0,255 | 0,341 | 0,326 | 0,317 | Newsg. |
| MapB | 0,479 | 0,199 | 0,255 | 0,291 | 0,242 | | | | |
| MapC | 0,413 | 0,240 | 0,268 | 0,300 | 0,257 | | | | |
| MapA | 0,408 | 0,133 | 0,193 | 0,242 | 0,243 | 0,179 | 0,215 | 0,300 | Reuters |
| MapB | 0,638 | 0,130 | 0,170 | 0,233 | 0,290 | | | | |
| MapC | 0,365 | 0,190 | 0,209 | 0,255 | 0,268 | | | | |
| k = 5 | | | | | | | | | |
| Map | Meta | | restrictive Base | | | Base | | | Dataset |
| | avg(loss) | avg(err) | base1 avg(err) | base2 avg(err) | base3 avg(err) | base1 avg(err) | base2 avg(err) | base3 avg(err) | |
| MapA | 0,704 | 0,346 | 0,400 | 0,441 | 0,376 | 0,506 | 0,470 | 0,559 | IMDB |
| MapB | 0,800 | 0,361 | 0,375 | 0,413 | 0,292 | | | | |
| MapC | 0,636 | 0,427 | 0,438 | 0,467 | 0,433 | | | | |
| MapA | 0,622 | 0,320 | 0,316 | 0,330 | 0,348 | 0,439 | 0,403 | 0,578 | Newsg. |
| MapB | 0,758 | 0,264 | 0,286 | 0,281 | 0,264 | | | | |
| MapC | 0,567 | 0,341 | 0,329 | 0,339 | 0,378 | | | | |
| MapA | 0,623 | 0,103 | 0,142 | 0,140 | 0,333 | 0,222 | 0,194 | 0,351 | Reuters |
| MapB | 0,735 | 0,111 | 0,136 | 0,111 | 0,290 | | | | |
| MapC | 0,571 | 0,150 | 0,155 | 0,163 | 0,351 | | | | |

Figure 1: Metaclustering Results for k=3 and k=5 on Reuters, Newsgroups and IMDB

| | k = 3 | | | k = 5 | | | Dataset |
|-----|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|----------------|
| | base1 avg(err) | base2 avg(err) | base2 avg(err) | base1 avg(err) | base2 avg(err) | base3 avg(err) | |
| 0.0 | 0,301 | 0,324 | 0,336 | 0,489 | 0,502 | 0,556 | IMDB |
| 0.1 | 0,289 | 0,316 | 0,319 | 0,486 | 0,487 | 0,540 | |
| 0.2 | 0,284 | 0,308 | 0,304 | 0,487 | 0,486 | 0,535 | |
| 0.3 | 0,279 | 0,302 | 0,284 | 0,485 | 0,488 | 0,528 | |
| 0.4 | 0,274 | 0,302 | 0,277 | 0,475 | 0,491 | 0,503 | |
| 0.5 | 0,260 | 0,295 | 0,260 | 0,465 | 0,478 | 0,466 | |
| 0.6 | 0,256 | 0,279 | 0,227 | 0,439 | 0,465 | 0,434 | |
| 0.7 | 0,244 | 0,269 | 0,213 | 0,416 | 0,447 | 0,378 | |
| 0.8 | 0,204 | 0,254 | 0,165 | 0,382 | 0,398 | 0,303 | |
| 0.9 | 0,178 | 0,199 | 0,098 | 0,329 | 0,365 | 0,213 | |
| 0.0 | 0,346 | 0,332 | 0,317 | 0,430 | 0,403 | 0,572 | Newsg. |
| 0.1 | 0,337 | 0,321 | 0,303 | 0,416 | 0,390 | 0,551 | |
| 0.2 | 0,325 | 0,311 | 0,293 | 0,403 | 0,380 | 0,521 | |
| 0.3 | 0,315 | 0,302 | 0,282 | 0,386 | 0,369 | 0,486 | |
| 0.4 | 0,302 | 0,294 | 0,272 | 0,371 | 0,363 | 0,451 | |
| 0.5 | 0,286 | 0,289 | 0,260 | 0,351 | 0,356 | 0,413 | |
| 0.6 | 0,266 | 0,278 | 0,229 | 0,328 | 0,347 | 0,365 | |
| 0.7 | 0,251 | 0,270 | 0,186 | 0,302 | 0,327 | 0,303 | |
| 0.8 | 0,237 | 0,259 | 0,142 | 0,270 | 0,294 | 0,234 | |
| 0.9 | 0,209 | 0,341 | 0,088 | 0,224 | 0,233 | 0,156 | |
| 0.0 | 0,219 | 0,221 | 0,292 | 0,211 | 0,205 | 0,348 | Reuters |
| 0.1 | 0,256 | 0,221 | 0,315 | 0,220 | 0,202 | 0,359 | |
| 0.2 | 0,246 | 0,242 | 0,317 | 0,210 | 0,191 | 0,358 | |
| 0.3 | 0,224 | 0,233 | 0,267 | 0,194 | 0,176 | 0,362 | |
| 0.4 | 0,207 | 0,218 | 0,258 | 0,170 | 0,171 | 0,342 | |
| 0.5 | 0,193 | 0,234 | 0,207 | 0,162 | 0,144 | 0,314 | |
| 0.6 | 0,175 | 0,214 | 0,191 | 0,171 | 0,163 | 0,291 | |
| 0.7 | 0,144 | 0,193 | 0,204 | 0,166 | 0,144 | 0,307 | |
| 0.8 | 0,176 | 0,162 | 0,181 | 0,138 | 0,132 | 0,269 | |
| 0.9 | 0,359 | 0,149 | 0,191 | 0,094 | 0,080 | 0,150 | |

Figure 2: Restrictive Base Methods for k = 3, k = 5 on Reuters, Newsgroups and IMDB

| Base Methods | | | Meta-Method | | Comb. Supervised Learning | | Dataset |
|---------------------|-------------------|-------------------|--------------------|----------|----------------------------------|------------------|----------------|
| base1 avg(err) | base2 avg(err) | base2 avg(err) | avg(loss) | avg(err) | svm avg(err) | tsvm avg(err) | |
| 0,246 | 0,255 | 0,231 | 0,409 | 0,159 | 0,228 | 0,204 | IMDB |
| 0,260 | 0,252 | 0,299 | 0,347 | 0,243 | 0,281 | 0,313 | Newsg. |
| 0,121 | 0,209 | 0,291 | 0,328 | 0,076 | 0,188 | 0,207 | Reuters |

Figure 3: Combination of Restrictive Clustering and Supervised Learning in Comparison with underlying Base Methods for k=2