Meta Methods for Model Sharing in Personal Information Systems

STEFAN SIERSDORFER
University of Sheffield, UK

SERGEJ SIZOV
University of Koblenz-Landau, Germany

This article introduces a methodology for automatically organizing document collections into thematic categories for Personal Information Management (PIM) through collaborative sharing of machine learning models in an efficient and privacy-preserving way. Our objective is to combine multiple independently learned models from several users to construct an advanced ensemble-based decision model by taking the knowledge of multiple users into account in a decentralized manner, e.g. in a peer-to-peer overlay network. High accuracy of the corresponding supervised (classification) and unsupervised (clustering) methods is achieved by restrictively leaving out uncertain documents rather than assigning them to inappropriate topics or clusters with low confidence. We introduce a formal probabilistic model for the resulting ensemble based meta methods and explain how it can be used for constructing estimators and for goal-oriented tuning. Comprehensive evaluation results on different reference data sets illustrate the viability of our approach.

Categories and Subject Descriptors: H.4.0 [**Information Systems Applications**]: General
General Terms: Algorithms, Theory, Experimentation
Computing Classification System Terms: Classification, Clustering, Peer-to-Peer
Additional Key Words and Phrases: personal information management, meta methods, restrictive methods

1. INTRODUCTION

The idea of a flexible environment that supports the user in producing, organizing, and browsing information originates in the early 1940s, a long time before the first personal computers and new communication tools like the Internet became available. The conceptual design of Vannevar Bush's memex [Bush 1945] (an acronym for Memory Extender) is probably the most cited (e.g. [Gemmell et al. 2002]) and criticized (e.g. [Buckland 1992]) representative of such early conceptual work. In his article, Bush described an integrated work environment that was electronically linked to a repository of microfilms and able to display stored contents and automatically follow references from one document to another. A number of visionary ideas from this early conceptual work can be recognized in state-of-the art information systems (cross-references between documents, browsing, keyword-based annotation of documents using the personal 'codebook', automatic generation of associative trails for content summarization, etc.).

Nowadays, the amount of data on our personal hard disks including emails, photos, web bookmarks, and documents grows exponentially [Gemmell et al. 2006], and improving computer support for personal information management (PIM) becomes more and more important. The work of the modern 'knowledge worker' [Haag et al.

2002] is characterized by complex human-centric processes [Kogan and Muller 2006] describing information intensive and complex tasks comprising of the creation, retrieval, digestion, filtering, and sharing of (large amounts of) personalized information. Although there are various approaches for desktop search [Dumais et al. 2003; google], for many tasks most users still prefer browsing their file system [Boardman and Sasse 2004; Teevan et al. 2004], which can be seen as navigating through a taxonomy consisting of categories and subcategories. The paradigm of a social semantic desktop (SSD) [Groza et al. 2007] aims to facilitate interoperation, data exchange, and collaboration between knowledge workers using Semantic Web technologies. The semantic desktop [Quan et al. 2003; Dong and Halevy 2005] can be understood as an integrated bundle of tools for search, browsing, problem-oriented custom information representation on the desktop, user profiling, and data analysis. By considering collaborative work and the interconnection of knowledge workers, recent social semantic desktops like Nepomuk [Groza et al. 2007] add support for the user's social environment, communication, and interaction.

Systems such as PHLAT [Cutrell et al. 2006] or Dogear [Millen et al. 2006] support *manual* tagging and the exchange of tags for personal data. But structuring the data manually into thematic categories is often very time-consuming for the user and not feasible. Traditionally, automatic methods based on machine learning paradigms (classification and clustering) have been used 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, focused crawling on the Web, and many more [Chakrabarti 2002]. In the past, PIM research considered machine learning methods in various frameworks and contexts. For instance, in TaskTracer [Dragunov et al. 2005] supervised learning using a previous chain of tasks, such as opening and saving files or visiting web pages, can be used for predicting subsequent tasks. In addition, there exists work on email classification (e.g., [Brutlag and Meek 2000; Segal and Kephart 1999; Klimt and Yang 2004]) and clustering [Surendran et al. 2005]. However, up to now, the aspect of automatic organization of personal data using *collaboration* has largely been neglected. For supervised learning, the main drawback is that a large amount of manually labeled data (training data) is required, which typically involves a considerable user effort.

In this article, we introduce a methodology for personal information organization in collaborative environments. In the context of a collaborative social environment that brings together multiple users with shared topics of interest, we explore whether it is possible to aggregate their knowledge and construct better machine learning models that can be used by every network member for their individual information demands. From a wider perspective, our methodology can be considered as an important building block for Nepomuk-like social semantic desktops [Groza et al. 2007] that allows for model-based collaborative organization and filtering of personal data.

One simple approach might be to share all available data, especially training samples, among all the users in the collaborative environment. However, the following reasons may prevent users from sharing all of their data with other members of the network:

—privacy, security, and copyright aspects of the user's personal information sources

—significantly increased network costs for downloads of additional training data on every peer

—increased runtime for the training of the decision models

We tackle this problem using *meta methods.* Our objective is to combine multiple independently learned classification models from several peers, and to construct an advanced decision model that simultaneously takes the knowledge of multiple users into account in a decentralized manner. A recent study [Millen et al. 2007] on a social tagging system shows that there is often a high degree of agreement between multiple users in the choice of categorical tags, and the research area of ontology alignment [Shvaiko and Euzenat 2005; Choi et al. 2006] deals with the problem of finding correspondences between categories in taxonomies of distinct users. We see the exchange of classification models for the underlying topics - the actual subject of this article - as the next logical step.

We assume that for PIM applications a high classification accuracy on a subset of the data is often more important than classifying all available data. To this end, we consider *restrictive* classifiers that either accept the document for a topic, or make no decision at all (i.e., abstain) if there is not sufficiently strong evidence. The latter option is important as it makes a key difference for constructing meta classifiers that combine the results of different models (e.g., in a quorum consensus manner). As a result, the quality metrics of interest are primarily the classification *error*, which is the fraction of incorrectly classified documents, and the document *loss*, which is the fraction of documents for which the classifier or meta classifier abstains.

In the classical scenario it is often assumed that all topic categories (classes) are known and that the training corpus provides example documents for all these categories. However in many real world applications these assumptions do not hold. We have to deal with the problem that the personal data of multiple users covers such a plethora (and growing number) of other topics that it is impossible to build a training set that comprises all these topics. A classifier trained to discriminate topics based on training data about "computer science", "mathematics", and "physics" is not trained to classify documents about, say, "esotericism", and assigns documents in an unstable way to one of the classes of the training documents; there is a significant difference between documents belonging to one of the training classes (classes of interest) and "junk" documents. We propose restrictive classification methods to tackle the "junk problem".

In many situations users might not be willing to create explicit training data, so that (unsupervised) clustering is the only viable option. We will show how the introduced concepts of model sharing, combination and restrictivity can be carried forward to the unsupervised scenario.

In the remainder of the article, we will focus on collaborative organization of personal documents; however, many of the techniques could be also applied to other media types.

Outline. The rest of this article is organized as follows. In Section 2 we describe related work from the area of machine learning and distributed learning. We describe the technical basics for our approach in Section 3. These include basic document representations and classification and clustering algorithms. Section 4 deals with restrictive classification and meta classification in more detail. We provide a probabilistic model for the tradeoff between *loss*, a measure for the restrictivity of a classifier, and the classification accuracy, and show how restrictive classification can be applied to eliminate “junk” documents. In Section 5, we carry the techniques used for restrictive meta classification forward to clustering. To this end, we introduce a technique to reduce the problem of combining cluster labels to combining class labels: the meta mapping. Finally, we apply the concepts of meta classification and clustering in the context of peer-to-peer information systems in Section 6. In Section 7 we show the result of systematic experiments for collaborative classification, clustering and junk elimination on various data sets. Section 8 provides a conclusion and a discussion of possible extensions of our framework in the context of personal information management.

2. RELATED WORK

There is a plethora of work on text document classification using a variety of probabilistic and discriminative models [Chakrabarti 2002]. The emphasis of this body of work has been on the mathematical and algorithmic aspects, and the engineering aspects of how to cope with tradeoffs and how to tune a classifier with regard to properties of the training data and, most importantly, specific application goals have been largely neglected (exceptions being, e.g., [Blok et al. 2001; Cronen-Townsend et al. 2002; Wang et al. 2003], which address different settings and are only marginally related to our work, however).

The machine learning literature has studied a variety of ensemble based meta methods such as bagging, stacking, or boosting [Breiman 1996; Wolpert 1992; Littlestone and Warmuth 1989; Freund 1999; Kuncheva 2004], and also combinations of heterogeneous learners (e.g., [Yu et al. 2002]). For bagging, an ensemble consists of classifiers built on bootstrap replicates of the training set. The classifiers outputs are combined by the plurality vote. For stacking, multiple classifiers are trained on parts of the training set and evaluated on the remaining training documents. The outputs of the classifiers are used as feature values for training a new classifier (stacked generalization). Boosting can be viewed as a model averaging method. Here a succession of models is built, each one trained on a data set in which the points misclassified by the previous model are given more weight. Our notion of a meta method is closest to bagging (see, e.g., [Breiman 1996]). To our knowledge none of the prior work on bagging and related techniques has considered the parameter tuning of such methods towards application-specific quality goals.

The approach of intentionally splitting a training set for meta learning has been investigated by [Chan 1996]. However, that work has focused on the efficiency versus accuracy tradeoff; so the improvements in efficiency were achieved at the expense of reduced accuracy. In contrast, our approach preserves and even improves high accuracy, and the measure that we are trading this for is document loss. The notion of loss in a ternary decision model, on the other hand, has not received wide attention. The paper [Schein et al. 2002] studied the accuracy-loss tradeoff in a ROC curve model (for a recommender system), but has not looked at how to systematically engineer and tune methods for judicious application choices regarding this tradeoff.

For SVM classifiers some isolated tuning issues have been considered in the literature. The popular SVM Light software package [Joachims 1998] provides various kinds of thresholds and variations of SVM training (e.g., SVM regression, transductive SVMs, etc.), but there is no systematic discussion of how to adjust these tuning knobs for a given application. [Brank et al. 2003] have proposed to introduce a bias for the separating hyperplane towards negative training samples, and advocated that this is beneficial when the number of positive training samples is very low.

Modifications of classifiers allowing for the rejection of test samples are described for Naive Bayes [Vailaya and Jain 2000] and SVM [Zhang and Metaxas 2006]. In [Fumera et al. 2003] restrictive classification is studied in the area of text categorization independent of our own work on restrictive classification of web documents [Siersdorfer and Sizov 2003]. A recent article on distance-based restrictive is [Landgrebe et al. 2006]. To our knowledge, these techniques were, up to now, not considered in the context of restrictive classification in a collaborative environment and junk reduction.

Label ranking [Brinker and Hüllermeier 2007; 2006] tackles the generalized classification problem of assigning an ordered list of class labels to a test document (instead of a single class label in the classical case). In that scenario, each training sample is accompanied by a list of class labels. For a given test case, the k nearest neighbors in the training set based on a similarity of distance measure are identified and their ordered lists combined. In contrast, we combine results produced by different classification and clustering models in order to take information about data from multiple sites into account.

There is work on combining multiple clustering methods in an ensemble learning manner, using consensus functions for clusterings based on information theoretic measures [Strehl and Gosh 2002], constructing a co-association matrix and performing hierarchical clustering on this matrix [Fred and Jain 2002], combining clusterings pair-wise and iteratively [Dimitriadou et al. 2002], using graph partitioning methods [Fern and Brodley 2004], or combining clusterings on different subspaces of a given feature space [Topchy et al. 2003]. None 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 estimation 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.

Algorithms for distributed clustering are described in [Kargupta et al. 2001; Li et al. 2003], but here document samples must be provided to a central server, making these solutions inconsistent with our requirements. The distributed execution of k-means was discussed in [Dhillon and Modha 2000]. However, this method requires multiple iterations that must be synchronized among the peers and causes a considerable amount of coordination overhead. Privacy-preserving distributed classification and clustering were also addressed in the prior literature: In [Vaidya and Clifton 2004] a distributed Naive Bayes classifier is computed; in [Merugu and

Ghosh 2003] the parameters of local generative models are transmitted to a central site and combined, but not in a P2P system. Another example of model exchange can be found in the TREC Spam Track [Cormack 2006] where participants exchanged their implementations of spam filters for evaluation purposes.

Techniques used in this article are based on our own work [Siersdorfer et al. 2004; Siersdorfer and Sizov 2004; Siersdorfer and Weikum 2005; Siersdorfer and Sizov 2006; 2007]. This article puts these concepts into the context of personal information management. We present a more unified and integrated view of the different methods, add novel theoretical concepts, and show new series of experiments.

3. TECHNICAL BASICS

3.1 Representation of Documents

All methods discussed in this paper represent documents as multidimensional feature vectors. In the prevalent bag-of-words model the features are derived from word occurrence frequencies [Baeza-Yates and Ribeiro-Neto 1999; Manning and Schuetze 1999] (e.g, capturing tf or $tf * idf$ weights of terms). In addition, feature selection algorithms [Madison et al. 1997] 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), information gain [Madison et al. 1997], or conditional MI [Wang and Lochovsky 2004]).

In the context of this paper, we primarily consider the standard bag-of-words approach [Baeza-Yates and Ribeiro-Neto 1999], including stopword elimination, stemming [Porter 1997], and tf based feature weighting with L1-normalization, without dimensionality reduction (i.e. without additional feature selection). This approach is consistently used for all evaluations in Section 7. However, our methodology can be combined with other document representations (e.g. $tf \cdot idf$ based features) as well; for this purpose, in [Goerlitz et al. 2008] we describe the infrastructure for maintaining accurate global statistics (e.g. idf) in a decentralized environment.

3.2 Text Categorization

Classifying text documents into thematic categories usually follows a supervised learning paradigm and is based on training documents that need to be provided for each topic. Feature vectors of topic labeled text documents (e.g., capturing $tf \cdot idf$ weights of terms) are used to train a classification model for each topic, using probabilistic (e.g., Naive Bayes) or discriminative models (e.g., SVM). Linear support vector machines (SVMs) 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. This training requires solving a quadratic optimization problem whose empirical performance is somewhere between quadratic and cubic in the number of training documents [Burges 1998]. 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. The decision simply requires computing a scalar product of the vectors $\vec{w}$ and $\vec{d}$. SVMs have been shown to perform very well for text classification (see, e.g., [Dumais et al. 1998; Joachims 1998]) and we use them as base classifiers for the classifier ensembles in our experiments in Section 7.

3.3 Unsupervised Partitioning of Document Collections

Clustering algorithms partition a set of objects, text documents in our case, into groups called *clusters*. They can be divided into the following groups [Ester et al. 2001]: partitioning methods, hierarchical methods, density based methods, grid based methods, and 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 [Han and Kamber 2001].

A simple, very popular member of the family of partitioning clustering methods is *k-Means* [Hartigan and Wong 1979]: 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. In our experiments in Section 7 we use k-Means as base clustering algorithms for our collaborative meta clustering methods. A similar algorithm, which can be considered as a "smoothed" form of k-Means is *EM clustering* [Han and Kamber 2001; Manning and Schuetze 1999]: in every iteration the probabilities of the objects for being contained in the different clusters are updated using the expectation-maximization technique.

4. RESTRICTIVE META CLASSIFICATION

4.1 Making Simple Classifiers Restrictive

The idea of restrictive classification is to avoid making a decision about a test document at all if that decision can be made only with relatively low confidence. So out of a given set of unlabeled data U, our method chooses a subset S of documents that are either accepted or rejected for the given topic label, and abstains on the documents in $U-S$. The quality measures precision, recall, F1, accuracy, and error are computed on the subset S, and we call the ratio $|U-S|/|U|$ the document *loss*.

We can use confidence measures to make simple methods restrictive. For SVMs a natural confidence measure is the distance of a test document vector from the separating hyperplane. 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 classifier restrictive by dismissing the L percent of the test documents with the lowest confidence values.

4.2 Restrictive Meta Classifiers

For meta classification we are given a set $V = \{v_1, \ldots, v_k\}$ of k binary classifiers with results $R(v_i, d)$ in $\{+1, -1\}$ for a document d, namely, $+1$ if d is accepted for the given topic by v_i, and -1 if d is rejected. We can combine these results into a meta result: $Meta(d) = Meta(R(v_1, d), \ldots, R(v_k, d))$ in $\{+1, -1, 0\}$ where 0 means abstention. A family of such meta methods is the linear classifier combination with thresholding [Siersdorfer and Sizov 2003]. Given thresholds t_1 and t_2, with $t_1 > t_2$, and weights $w(v_i)$ for the k underlying classifiers we compute $Meta(d)$ as follows:

$$Meta(d) = \begin{cases} +1 & \text{if } \sum_{i=1}^{k} R(v_i, d) \cdot w(v_i) > t_1 \\ -1 & \text{if } \sum_{i=1}^{k} R(v_i, d) \cdot w(v_i) < t_2 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

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

(1) voting [Breiman 1996]: Meta returns the result of the majority of the classifiers (parameterization: $w(v_i) = 1$ for all $v_i \in V, \quad t_1 = t_2 = 0$).

(2) unanimous decision: if all classifier give us the same result (either +1 or -1), Meta returns this result, 0 otherwise.
(parameterization: $w(v_i) = 1$ for all $v_i \in V, \quad t_1 = k - 0.5 = -t_2$)

(3) weighted averaging [Wang et al. 2003]: Meta weighs the classifiers by using some predetermined quality estimator, e.g., a leave-one-out estimator for each v_i.

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. (For SVM outputs, scores can be mapped to probabilities as described, e.g., in [Platt 1999], which could be used for further, class-specific, adjustments of the thresholds.)

The approach itself carries over to more sophisticated instantiations of the meta classifier framework. It can be extended, e.g., by allowing the abstain option for the classifiers v_i, i.e. $R(v_i, d) \in \{+1, -1, 0\}$. This would introduce an additional stage of restrictivity, and could make sense in situations where restrictive base classifiers $R(v_i, d)$ with specifically good loss-error behavior can be constructed. Instead of hard class assignments $R(v_i, d)$, we can also use soft assignment values $Res(v_i, d) \in [-1, 1]$ (e.g., for SVM based on normalized distances of the test points from the separating hyperplane); we use this approach for our experiments in Section 7.

Another extension is the multi-classification scenario, i.e. having $l > 2$ classes $\{1, \ldots, l\}$. The meta classification framework can be generalized by computing for each class j a weighted sum

$$\sum_{i=1}^{n} \Delta_j(R(v_i, d)) \cdot w(v_i) \tag{2}$$

(where $(R(v_i, d) \in \{1, \ldots, l\}$, and $\Delta_j(i) = 1$ if $i = j$ and -1 otherwise), and, analogously to the ternary case, introducing restrictivity by defining thresholds $\{t_1, \ldots, t_l\}$. This can be applied in a natural way to classifiers for multi-classification such as kNN, Naive Bayes, or variants of Rocchio Classifiers; for most of these classifiers the outputs can be interpreted as confidence values. In many cases, the multi-class problem is also reduced to multiple binary classification problems that can be solved separately [Allwein et al. 2001; Masulli and Valentini 2000] and there are various possibilities to obtain confidence values. (For instance, for binary SVM we can consider the largest distance of a test document from the hyperplane separating the class, selected by any combination method for multi-classification, from the other classes.) Scenarios where the underlying binary classifiers are allowed to abstain are possible as well. In this case, a simple option might be to abstain for the multi-class case if a certain fraction of the underlying binary classifiers abstains for a given class. In the rest of the work we will consider only the binary (or actually, because of the abstention option, ternary) unanimous-decision meta classifier as the simplest of the above cases in order to demonstrate the feasibility of our approach.

4.3 Tradeoffs for Restrictive Classification

In this section we describe the tradeoffs that occur in restrictive classification if the test set contains additional junk documents (i.e., documents not belonging to classes the classifier was trained to separate). Consider a training set T consisting of documents from two classes $class_a$ and $class_b$, and a set of unlabeled documents U containing documents from $class_a$ and $class_b$, and *junk* documents that belong to neither of these classes. (The scenario can be easily generalized to a set of l classes $C = \{c_1, \ldots, c_l\}$ instead of two classes.) Given a document $d \in U$, a restrictive classifier gives us the result A if it classifies the document into $class_a$, B if it classifies the document into $class_b$, 0 if the classifier abstains. The possible combinations between the real classes and the possible results of a classifier are shown in the contingency table in Figure 1. In this notation BA is the set of documents in $class_b$ which are assigned to class $class_a$ by the classifier, $J0$ is the set of junk documents from U where the classifier abstains, etc.

		classification		
		A	B	0
real class	$class_a$	AA	AB	A0
	$class_b$	BA	BB	B0
	junk	JA	JB	J0

Fig. 1. Contingency Table for Restrictive Classification with Junk Reduction

An appropriate restrictive classifier should optimize the following quality measures:

(1) Maximize *junk reduction* (fraction of junk documents dismissed by the classifier):

$$junkRed := \frac{|J0|}{|JA| + |JB| + |J0|} \tag{3}$$

(2) Minimize *loss* (fraction of dismissed documents from the classes of interest $class_a$ and $class_b$):

$$loss := \frac{|A0| + |B0|}{|AB| + |AA| + |BA| + |BB| + |A0| + |B0|} \tag{4}$$

(3) Minimize *error* (fraction of non-dismissed documents classified into the wrong class):

$$error := \frac{|AB| + |BA| + |JA| + |JB|}{|AA| + |AB| + |BA| + |BB| + |JA| + |JB|} \tag{5}$$

(or maximize $accuracy = 1 - error$)

As *document reduction* (not to confuse with the loss), we define the fraction of documents in U, where the classifier abstains:

$$docRed := \frac{|A0| + |B0| + |J0|}{|U|} \tag{6}$$

The document reduction can be observed directly from the classifier output without knowing the real class labels of the documents in U. The document reduction has an implicit influence on *junkRed*, *loss* and *error*. Note that in the special case that the set does not contain junk documents, $|JO| = |JA| = |JB| = 0$, *docRed* = *loss*, and *error* is the fraction of incorrectly classified documents as described before. We are then just dealing with a more simple *loss*-*error* tradeoff.

4.4 Using Restrictive Meta Classification for Junk Elimination

Up to now it was assumed that all underlying classifiers had sufficient training data: samples for every thematic that might occur among the test documents. In this section we drop this assumption and make a major step forward to cope with corpora that are not necessarily "in tune" with the thematic classes that were defined apriori. This is a very significant case with "open" corpora like the Web with a huge amount of topics and documents for which comprehensive training is absolutely impossible [Siersdorfer and Weikum 2005]. Classifiers trained to discriminate topics based on training data about "computer science", "mathematics", and "physics" are not trained to classify documents about, say, "esotericism ", and assign documents in an unstable way to one of the classes of the training documents; there is a significant difference between documents belonging to one of the training classes (classes of interest) and "junk" documents.

In practice we observe a tradeoff between the quality measures defined above: there is a tradeoff between the loss on one hand and junk-reduction and error on the other hand. We can use restrictive methods and meta methods to make simple methods restrictive, as described in Section 4.1 and Section 4.2.

A Probabilistic Model for Restrictive Meta Methods in a Junk Reduction Scenario. In this section we present a probabilistic model for the case that test documents may contain junk documents and we provide approximations for *loss*, *error* and *junkRed*. This leads to a better understanding of why meta classification can be used for junk reduction.

Consider the unanimous-decision meta method. We associate a Bernoulli random variable X_i with each classification method v_i, where $X_i = 1$ if v_i classifies a document into $class_a$ and $X_i = 0$ if v_i classifies a document into class $class_b$. We want to compute the probability $P(X_1 = \ldots = X_k|Junk)$ that the classifiers v_i provide a unanimous decision if they are presented a junk document. From basic probability theory it follows that

$$P(X_1 = 1 \wedge X_2 = 1|Junk) = \\ cov(X_1, X_2|Junk) + P(X_1 = 1|Junk) \cdot P(X_2 = 1|Junk) \quad (7)$$

where

$$cov(X_1, X_2|Junk) = \frac{1}{n-1} \sum_j (x_1 - \overline{x_1})(x_2 - \overline{x_2}) \quad (8)$$

is the covariance for the data points $(x_1, x_2) \in \{0,1\} \times \{0,1\}$ (with mean values $\overline{x_1}$ and $\overline{x_2}$) of the joint distribution of (X_1, X_2) on the set of junk documents. To model the most important correlations among $l > 2$ classification methods we use a tree dependence model, which is a well known approximation method in probabilistic IR ([Van Rijsbergen 1977]). We define a *Dependence Graph* $G = (V, E)$ where V consists of the Bernoulli variables X_i, and where E is the set of undirected edges $e(X_i, X_j)$ with weight $w(e(X_i, X_j))) = cov(X_i, X_j)$ for all X_i,X_j $(i \neq j)$. We approximate the Dependence Graph by a maximum spanning tree $G' = (V, E')$ - with nodes V and edges 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:

$$P(X_1 = x_1, \ldots, X_k = x_k|Junk) = \\ P(X_{root} = x_{root}|Junk) \prod_{(i,j) \in E'} \frac{P(X_i = x_i, X_j = x_j|Junk)}{P(X_i = x_j|Junk)} \quad (9)$$

where X_{root} is the root node of the tree G' and $x_i \in \{0,1\}$. Let $w(e(X_i, X_j)) = cov_{ij}$ the covariances of the pairs of adjacent classifiers v_i, v_j. Then we have:

$$P(X_1 = x_1, \ldots, X_k = x_k|Junk) = \\ P(X_{root} = x_{root}|Junk) \prod_{(i,j) \in E'} \frac{P(X_i = x_i|Junk)P(X_j = x_j|Junk) + cov_{ij}}{P(X_i = x_j|Junk)} \quad (10)$$

and

$$P(X_1 = 1, \ldots, X_k = 1|Junk) = \\ P(X_{root} = 1|Junk) \prod_{(i,j) \in E'} \frac{P(X_i = 1|Junk)P(X_j = 1|Junk) + cov_{ij}}{P(X_i = 1|Junk)} \quad (11)$$

Analogously we obtain $P(X_1 = 0, \ldots, X_k = 0|Junk)$. To compute the probabilities that all classifiers v_i classify a document into the same class, if the document belongs to one of the classes in $C = \{class_a, class_b\}$, we associate a Bernoulli variable X'_i with each classification method v_i, where $X'_i = 1$ if v_i classifies a document correctly, 0 otherwise. We want to compute the probabilities $P(X'_1 = 1, \ldots, X'_k = 1|C)$ and $P(X'_1 = 0, \ldots, X'_k = 0|C)$ that all classifiers classify a document correctly / incorrectly if the document belongs to one of the classes in C. With analogous arguments as above we obtain the following approximation:

$$P(X'_1 = 1, \ldots, X'_k = 1|C) = \\ P(X'_{root} = 1|C) \prod_{(i,j) \in E''} \frac{P(X'_i = 1|C)P(X'_j = 1|C) + cov'_{ij}}{P(X'_i = 1|C)} \quad (12)$$

where cov'_{ij} are the covariances on the documents in C. Analogously we obtain $P(X'_1 = 0, \ldots, X'_k = 0|C)$.

Let $P(C)$ be the probability that a document belongs to a class in C and $P(Junk)$ be the probability that a document is a junk document. Then we obtain approximations for *junkRed*, *loss*, *error*, and *docRed* by inserting the above expressions into:

$$junkRed = 1 - (P(X_1 = 1, \ldots, X_k = 1|Junk) + P(X_1 = 0, \ldots, X_k = 0|Junk)) \tag{13}$$

$$loss = 1 - (P(X'_1 = 1, \ldots, X'_k = 1|C) + P(X'_1 = 0, \ldots, X'_k = 0|C)) \tag{14}$$

$$error = \frac{P(C)P(X'_1 = 0, \ldots, X'_k = 0|C) + P(Junk)P(X_1 = \ldots = X_k|Junk)}{1 - junkRed \cdot P(Junk) - loss \cdot P(C)} \tag{15}$$

$$docRed = junkRed \cdot P(Junk) + loss \cdot P(C) \tag{16}$$

We assume that a classifier v_i behaves less stable on *junk* documents than on documents from classes of interest $class_a$ and $class_b$. This means that the tendency of a classifier to classify a document into the correct class (either $class_a$ or $class_b$) for non-junk documents is higher than its tendency to classify junk documents into a specific class $class_a$ or $class_b$. Formally, this assumption can be described by:

$$|P(X_i = 1|Junk) - 0.5| < |P(X'_i = 1|C) - 0.5| \tag{17}$$

(Note that for the considered the binary case, 0.5 is the probability of correct random classification. Furthermore, note that equation 17 also holds if we replace $P(X_i = 1|Junk)$ with $P(X_i = 0|Junk)$ or $P(X'_i = 1|C)$ with $P(X'_i = 0|C)$.) The instability of a classifier when presented with *junk* documents results in a higher disagreement among the members of the classifier ensemble, and, thus, a higher dismission rate for junk documents than for documents of interest (i.e. higher junk reduction *junkRed* and lower *loss*).

As an illustrative example we consider the case that the $k > 2$ classification methods have the same probability $p < 0.5$ to assign a document from C (i.e. the classification methods perform better than random), that we have in all cases a covariance $c < p(1-p)$ (i.e. the classification methods are not perfectly correlated.), and that our document corpus contains a fraction $0 < j < 1$ of junk documents. Let furthermore $q := P(X_i = 1|Junk)$ with the property $|q - 0.5| < |p - 0.5|$ described in equation 17. In this case we would obtain for *junkRed*, *loss* and *error*:

$$junkRed = 1 - \left((1-q)\left(\frac{c + (1-q)^2}{1-q}\right)^{k-1} + q\left(\frac{c + q^2}{q}\right)^{k-1}\right) \tag{18}$$

$$loss = 1 - \left((1-p)\left(\frac{c + (1-p)^2}{1-p}\right)^{k-1} + p\left(\frac{c + p^2}{p}\right)^{k-1}\right) \tag{19}$$

$$error = \frac{(1-j)p\left(\frac{c+p^2}{p}\right)^{k-1} + j(1-q)\left(\frac{c+(1-q)^2}{1-q}\right)^{k-1} + jq\left(\frac{c+q^2}{q}\right)^{k-1}}{j(1-q)\left(\frac{c+(1-q)^2}{1-q}\right)^{k-1} + jq\left(\frac{c+q^2}{q}\right)^{k-1} + (j-1)(1-p)\left(\frac{c+(1-p)^2}{1-p}\right)^{k-1} + (j-1)p\left(\frac{c+p^2}{p}\right)^{k-1}} \tag{20}$$

It is easy to show that for $k \to \infty$ the loss converges monotonically to 1, and the error to 0 (i.e. with more classification methods we can obtain a lower error but pay the price of a higher loss). Furthermore also $junkRed$ converges to 1 and the salient invariant $loss > junkRed$ holds. Even $\frac{1-loss}{1-junkRed}$ converges to ∞; this means, that with increasing k we dismiss much more junk documents than documents of interest. The covariance plays the role of a "smoothing constant": with higher correlated classification methods the convergence of both loss and error is slowed down. Note, that the conditions in the example can be further generalized. For instance, we can drop the assumption that all covariances c are equal. It would be sufficient that the covariances c_{ij} of random variables X_i, X_j (behavior on junk) are smaller or equal than the covariances c'_{ij} of random variables X'_i, X'_j (behavior on non-junk documents), i.e., if pairs of classifiers behave 'more independently' on junk documents. Further relaxations for the covariances are possible if we make stronger assumptions about probabilities p and q.

5. RESTRICTIVE META CLUSTERING

This section addresses the problem of automatically structuring heterogeneous document collections by using clustering methods. We adapt and extend the concept of restrictive (supervised) classification, described above to (unsupervised) clustering. So, 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, and better overall accuracy, and make unsupervised self-organization more robust. Note that in the considered unsupervised scenario, we do not have any pre-defined classes or training data; thus, our notion of "junk" cannot be directly carried forward to clustering. For the unsupervised scenario described in this section, restrictivity is only used to tune the error - loss tradeoff.

5.1 Making Simple Methods Restrictive

Analogously to the idea of restrictive classification, 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 $U - S$. Analogously to the supervised case, 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 inverse distance of a document vector from the nearest centroid (or some other similarity measure). So we can tune these methods by requiring that the documents accepted for one of the clusters have a distance below 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.

We note that the idea of dismissing data points for a clustering has been considered before in the context of noise and outlier detection. For instance, in [Davé 1991] an additional noise cluster is introduced and iteratively updated using a fuzzy version of the k-means algorithm. In our work we are rather focused on the experimentally observed *tradeoff* between cluster quality and loss.

5.2 Restrictive Meta Methods

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

5.2.1 *Meta mapping.* To combine the $c_i(d)$ into a meta result, 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 with each other which are "most correlated".

Formally, for every method c_i we want to determine a bijective function $map_i : \{1, \ldots, k\} \rightarrow \{1, \ldots, k\}$ which assigns each label $a \in \{1, \ldots, k\}$ assigned by c_i to a meta label $map_i(a)$. By this mapping 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:

$$res_i(d) := map_i(c_i(d)) \tag{21}$$

How can we obtain the map_i functions? One method is to maximize the correlation between the cluster labels. For sets $A_1, \ldots, A_x$, we can define their *overlap* as

$$overlap(A_1, \ldots, A_x) := \frac{|A_1 \cap \ldots \cap A_x|}{|A_1| + \ldots + |A_x| - |A_1 \cap \ldots \cap A_x|} \tag{22}$$

Now using

$$A_{ij} := \{d \in U | res_i(d) = j\} \tag{23}$$

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

$$\frac{1}{k} \sum_{j=1}^{k} \frac{1}{\binom{l}{2}} \sum_{(i,m) \in \{1,\ldots,k\}^2, i<m} overlap(A_{ij}, A_{mj}) \tag{24}$$

We are interested in the mappings map_i which maximize the average overlap. However there is a combinatory explosion: there are $k!^{l-1}$ possibilities to build mappings. This problem can be transformed into a multi-dimensional assignment problem (MAP) [Pierskalla 1968] which has been shown to be NP-complete; thus this approach is only viable for small values of k and l. 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_{l-1} and c_l, and to use transitivity to compute an overall mapping. Each of these subproblems can be formulated as an assignment problem that can be solved by the Hungarian Algorithm [Kuhn 1955] which has been shown to have a runtime complexity of $O(k^3)$. Since we have to solve $l-1$ of such subproblems, the complexity of the greedy approach is $O(k^3 \cdot l)$. 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. This can be done by first sorting the k^2 overlap measures in time $O(k^2 \log k^2) = O(k^2 \log k)$, and at most one scan of the obtained list. Thus the overall complexity is $O(k^2 \log k \cdot l)$.

Two alternative mapping approaches are based on prioritizing clusters that produce a high overlap in one particular combination and low overlaps otherwise (i.e. taking the variance of the overlaps into account), and on association rule mining (considering tuples of cluster methods and labels as items, and generating associations between these items). A detailed description of these approaches can be found in our paper [Siersdorfer and Sizov 2004].

5.2.2 *Meta Functions.* After having computed the mapping we are given a set $C = \{c_1, \ldots, c_l\}$ of l 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: $Meta(d) = Meta(res_1(d), \ldots, res_l(d))$ in $\{+1, -1, 0\}$ where 0 means abstention. This is analogous to combination of results in meta classification, as described in Section 4.2. 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:

$$Meta(d) = \begin{cases} +1 & \text{if } \sum_{i=1}^{l} res_i(d) \cdot w(c_i) > t_1 \\ -1 & \text{if } \sum_{i=1}^{l} res_i(d) \cdot w(c_i) < t_2 \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

These considerations can be easily extended to the case of $k >= 2$ possible clusters, e.g., by computing the linear combination for each cluster label separately, and by assigning the label with the maximum value to the document d if this value is above some threshold.

Analogously to meta classification, the restrictive and tunable behavior is achieved by the choice of the thresholds as t_1 and t_2. For this family of meta methods we obtain the same special cases, as described in Section 4.2 for classification and this leads to a probabilistic model similar to the model for classification (see [Siersdorfer and Sizov 2004] for more details). Furthermore, extensions such as restrictive base method and clusterings with $k > 2$ partitions can be obtained in the same way as in the meta classification scenario described in Section 4.2 (dealing now with cluster meta labels $res_i(d) = map_i(c_i(d))$ instead of class labels $R(v_i, d)$); i.e., considering $k > 2$ classes $\{1, \ldots, k\}$, the meta clustering framework can be generalized by computing for each cluster j a weighted sum

$$\sum_{i=1}^{l} \Delta_j(res_i(d)) \cdot w(c_i) \tag{26}$$

(where $res_i(d) \in \{1, \ldots, k\}$, and $\Delta_j(i) = 1$ if $i = j$ and -1 otherwise), and, analogously to the ternary case, introducing restrictivity by defining thresholds $\{t_1, \ldots, t_k\}$.

5.3 Meta Clustering using Confidence Values for Clustering

Up to now, we have assumed that each document is assigned to exactly one cluster. We drop this assumption now, and consider "soft" assignments of cluster labels. For example EM clustering [Manning and Schuetze 1999; Han and Kamber 2001] assigns to a document the probabilities of membership to the different clusters. Confidence values for cluster memberships can be also assigned to the results of other clustering algorithms such as k-means which we use in our experiments (e.g. by taking normalized similarities to the centroids representing the clusters into account).

5.3.1 *Generalized Meta Clustering Problem.* For meta clustering we are given a set $C = \{c_1, \ldots, c_l\}$ of clustering methods, and a set U of unlabeled documents. Each method c_i assigns to a cluster label j in $\{1, \ldots, k\}$ and to a document d in U a cluster confidence:

$$c_i(j, d) = c_{ij}(d) \tag{27}$$

with

$$\sum_{j=1}^{k} c_{ij}(d) = 1 \text{ for all } i = 1, \ldots, l \tag{28}$$

Our objective is to find l meta mappings

$$map_i : \{1, \ldots, k\} \rightarrow \{1, \ldots, k\} \tag{29}$$

that map the cluster labels of each method c_i to a meta label. Furthermore we aim to determine a meta function

$$meta(d) = meta(c_{11}(d), \ldots, c_{lk}(d), map_1, \ldots, map_l) \tag{30}$$

that assigns to each document d a meta label in $\{1, \ldots, k\} \cup \{0\}$ ("0" means abstention, as described above).

5.3.2 *Meta Mapping.* Now we generalize the correlation-based mapping, described in Section 5.2.1, for the occurrence of soft assignments. We define:

$$|A_{ij}| = \sum_{d \in U} c_{ix}(d) \tag{31}$$

with

$$x = map_i^{-1}(j) \tag{32}$$

We define the intersection, $|A_{ij} \cap A_{mj}|$, as follows:

$$|A_{ij} \cap A_{mj}| = \sum_{d \in U} c_{ix}(d) \cdot c_{my}(d) \tag{33}$$

with

$$x = map_i^{-1}(j) \text{ and } y = map_m^{-1}(j) \tag{34}$$

Note that for the special case of a "hard" assignment of documents to clusters, i.e,

$$\begin{aligned} c_{ij}(d) = 1 \text{ for exactly one } j \in \{1, \ldots, k\}, 0 \text{ otherwise} \\ \text{for all } i \in \{1, \ldots, l\}, d \in U \ , \end{aligned} \tag{35}$$

these definitions are equivalent to those given in Section 5.2.1. Now we can define the optimization problem exactly as in Section 5.2.1. For the overlap we have:

$$overlap(A, B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{36}$$

We obtain the mapping by solving the following optimization problem:

$$\begin{aligned} &\text{Maximize over } (map_1, \ldots, map_l) : \\ &\frac{1}{k} \sum_{j=1}^{k} \frac{1}{\binom{l}{2}} \sum_{(i,m) \in \{1,\ldots,k\}^2, i<m} overlap(A_{ij}, A_{mj}) \end{aligned} \tag{37}$$

5.3.3 *Meta Functions.* Having obtained the mapping map_i as described above, we can compute the meta labels for clustering method c_i applied to document d as:

$$res_i(d) = map_i(\arg\max_x c_{ix}(d)) \text{ for all } i \in \{1, \ldots, l\} \tag{38}$$

and compute the meta result by a linear combination of these results as described in Section 5.2.2. Alternatively, we can take the confidence values not just for the mapping, but also for the meta function into account, for instance, by replacing the $res_i(d)$ with the confidence values $\max_x c_{ix}(d)$.

6. RESTRICTIVE METHODS AND META METHODS IN PEER-TO-PEER SYSTEMS

In this Section we apply the meta classification and clustering approach, described above, in the context of peer-to-peer (P2P) systems. Our approach is to combine models from multiple peers and to construct an advanced decision model that takes the knowledge of multiple P2P users into account.

6.1 System Architecture

The conceptual infrastructure of a peer in our distributed setting consists of two layers [Siersdorfer and Sizov 2006]. The *lower (infrastructure) layer* determines the communication among the peers. We assume that all peers share the same thematic taxonomy such as *dmoz.org* [dmoz]. The *upper (meta modeling) layer* is the distributed algorithm that utilizes results from particular peers to construct improved learning models (e.g. classification and/or clustering models) that can be used to organize the users' personal data, and to adjust the topics of a user-specific personalized ontology.

6.2 Organization of the Infrastructure Layer

As an example, we consider the integration of our approach with Minerva [Bender et al. 2004], [Bender et al. 2005], a decentralized Web search engine that allows handling large amounts of Web data in a distributed and decentralized manner. Every peer is autonomous and has a local index that can be built from peer's own, locally stored data (e.g. results of the thematically focused Web crawl). The index contains inverted lists with Web documents that are associated with particular topics of peer's interest. The distributed directory is layered on top of a Chord dynamic hash table (DHT) [Stoica et al. 2001] which holds compact, aggregated information about the peer's local collections. The DHT is used for partitioning the space of existing topics of interest: every peer is responsible for a certain subset within the global environment. Thus, peers in our framework play two roles: directory peers and index peers.

Directory maintenance and location of relevant peers is organized as follows. Every peer publishes a summary about its topics of interest to the global directory. A default hash function of the DHT is applied to the topic label in order to determine the directory peer currently responsible for this topic. Posts contain contact information of the peer (i.e. its IP address, port, availability, etc.) as well as peer's collection statistics for calculating IR relevance measures. The statistics include the cardinality of the topic data set and the compact representation of its content in form of a compressed Bloom filter [Bloom 1970], [Bender et al. 2005].

The directory peer maintains an index list of all postings for the given topic label, which contains addresses of peers that provide data collections on the given topic, and received statistics about them. The statistics are used to support the peer selection process for a given collaborative learning task. The location of relevant peers for collaborative data organization task (e.g. building of classification meta models for a given topic pair) proceeds as follows. Using the underlying overlay network directory, the initiator of the meta model construction retrieves a list of potentially relevant peers for each topic by querying corresponding index peers over the DHT. The lookups are performed separately for each topic label. Responsible index peers return lists of potentially relevant nodes, along with their Bloom filter based content summaries. The intersection of received lists contains candidate peers that own training data for *all* desired topics and thus can contribute to the meta model construction.

Using the database selection methods from distributed IR, a (smaller) subset of promising contributors for meta model construction is identified. We adopt the peer selection methodology from the Minerva system [Bender et al. 2005] which is based on the overlap estimator between data collections by comparing their Bloom filters. Our objective for meta model construction is to utilize training collections with possibly low overlap; for this reason, we incrementally select from the candidate list peers with highest estimated novelty [Bender et al. 2005] of training set until the desired number of meta model contributors is reached. To further increase the lookup efficiency, frequently used pairs of topic labels can be separately indexed in the DHT as described in [Bender et al. 2006].

Subsequently, the request to join the meta model construction is forwarded to the chosen peers. The peers locally construct decision models for requested topics using their own, locally stored data sets, and exchange these models in pairwise point-to-point manner, allowing for efficient communication and limiting the load of the global directory. Finally, each peer combines the received models into a meta model and starts using it for local data organization.

Maintenance of the distributed directory, location of resources, and selection of most relevant peer collections are common tasks for P2P information systems and search engines. Insofar, for locating peers with suitable topic-specific collections we can exploit existing systems like Minerva [Bender et al. 2005] without adding any further storage/communication overhead to its infrastructure. In the next step, we focus our analysis on more specific issues for our approach, namely, meta model and estimator construction in a decentralized setting.

6.3 Exchanging Data Models among Peers

In our framework we are given a set of k peers $P = \{p_1, \dots, p_k\}$. Each peer p_i maintains its collection of documents D_i. The idea is to build concise individual models on each peer and then combine the models into a meta model. More formally, in the first step each peer p_i builds a model $m_i(D_i)$ using its own document set D_i. In the second step, the models m_i are propagated among the k peers as described in Section 6.2. To avoid high network load, it is crucial for this step that the models m_i are a very compressed representation of the document sets D_i. In the next step, each peer p_i uses the set of received models $M = \{m_1, \dots, m_k\}$ to construct a meta model $Meta_i(m_1, \dots, m_k)$. From now on, p_i can use the new meta model $Meta_i$ (instead of the 'local' model m_i) to analyze its own data D_i.

For classification, instead of transferring the whole training sets T_i, only the models m_i need to be exchanged among the peers. For instance, linear support vector machines (SVMs), as described in Section 3, can be represented in a very compressed way: as tuples $(\vec{w}, \vec{l}, b)$ of the normal vector $\vec{w}$ and bias b of the hyperplane and $\vec{l}$, a vector consisting of the encodings of the terms (e.g., some hash code) corresponding to the dimensions of $\vec{w}$. The ith component of vector l relates the ith component of w to its corresponding feature. Thus, $\vec{l}$ provides us with a synchronization between the feature spaces of the different peers. Each peer combines the received classification models to a meta model as described in Section 4.

Similar space saving representations are possible for other learning methods (e.g., Bayesian Learners). In addition, building the classifiers this way is much more efficient than building one 'global' classifier based on $T = \bigcup T_i$ because the computation is distributed among the peers, and for classifiers with highly nonlinear training time (e.g. SVM) the splitting can save a lot of time (see [Siersdorfer et al. 2004]).

Analogous representations can be obtained for clustering models. For the k-means clustering algorithm (see Section 3), the clustering model can be represented as $(\vec{z_1}, \dots, \vec{z_l}, \vec{l})$, where the $\vec{z_i}$ are vector representations of the computed centroids, and $\vec{l}$ contains encodings of the feature dimensions, and provides us with a synchronization of the feature spaces, as described above for classification. Note that exchange of clustering models is not restricted to k-means. For algorithms and data resulting in non-spherical clusters, e.g., cluster boundaries might be a better representation. Furthermore, a clustering on a given peer using any arbitrary partitioning method can be transformed into a classification model (e.g. based on SVMs) by using the clustered data points along with their cluster labels as training samples. On a given peer the clustering models are first applied to its data, and the results are combined using meta clustering, thus, allowing for the combination of different clustering algorithms. The same holds for meta classification; thus it is possible to combine different classification methods from multiple peers. We can then exchange the clustering or classification models and combine them using meta methods as described in Section 4.2 and 5

We note that from the exchanged classification and clustering models some information about the word distribution of the user's documents can be inferred. The exchange of a certain amount of user information is unavoidable for collaboration. However the models can be seen as a very compressed statistical representation of the data, and, thus, reveal much less information than the full content.

6.4 Estimators and Tuning

For a restrictive meta classifier, we are interested in its behavior in terms of *error* and *loss* (fraction of unclassified documents). A typical scenario could be a number of users in different peers accepting a loss up to a fixed bound, to obtain a lower classification error for the remaining documents. In our prior work [Siersdorfer et al. 2004] the tuning of the number k of classifiers for a user-acceptable loss threshold was described. We will not repeat this here and will instead focus on the P2P specific aspects. The main ingredients of the estimation and tuning process are:

(1) estimators for base classifiers (based on cross-validation between the training subsets T_i)

(2) estimators for the pairwise correlations between the base classifiers $\{m_1, \ldots, m_k\}$

(3) probabilistic estimators for loss and error based on 1. and 2.

For the cross-validation, at least two peers, p_i and p_j, must cooperate: p_i sends a tuple $(m_i, IDs(T_i))$, consisting of its classifier m_i and a list of IDs (not contents!) of its training documents, to p_j. The peer p_j uses the list of submitted IDs to identify duplicates in both collections and performs cross-validation by m_i on $T_j - T_i$. (In the Web context, the IDs of T_i can be easily obtained by computing content-based 'fingerprints' or 'message digests' (e.g. MD5 [Rivest 1992])). The resulting error estimator (a simple numerical value) for m_i can be forwarded from p_j back to p_i or to other peers. For the computation of pairwise covariance, at least three peers, p_i,p_j and p_m, must cooperate: p_i and p_j send their classifiers and document IDs to p_m and p_m cross-validates in parallel both classifiers on $T_m - T_i - T_j$. By this procedure we get also accuracy estimators.

Finally, the estimators for *covariance* and *accuracy* (numerical values) can be distributed among the peers and estimators for the overall meta classifier can be built. When the estimated quality of the resulting meta classifier does not meet the application-specific peer requirements (e.g. the expected accuracy is still below the specified threshold), the initiating peer may decide to invoke additional nodes for better meta classification. Note that for meta clustering, estimators *cannot* be built in the same easy way, because for the unsupervised case we cannot evaluate base methods by cross-validation.

6.5 Complexity

To illustrate the advantages of the presented methodology, we consider a simplified cost analysis for the meta model construction in a decentralized setting. We primarily focus on communication costs, as storage and local model training are rather uncritical.

In line with experiments presented in Section 7 we assume that $P = 16$ peers from a large-scale overlay network cooperate in order to construct the meta classifier. The arbitrary training collection from Section 7 (del.icio.us experiments) for one topic pair consists on average of 10 Mb of compressed data. In contrast, the size of the resulting linear SVM classifier after compression is ca. 1 Kb. Consequently, the overall effective network payload (excluding protocol overhead) of the combinational meta method is less than 250 Kb. For the same setting, the naive point-to-point exchanging of training collections between peers would cause the total effective payload of ca. 2.4 Gb. The previously discussed alternate 'superpeer-solution' (i.e. one peer collects all training sets, constructs the meta classifier, and distributes it to all invoked peers) would still cause the payload greater 150 Mb. The further crucial drawback of the two latter solutions is the growing computational overhead for learning on larger training collections, which can become prohibitively high for methods with nonlinear training complexity.

The difference rapidly increases with higher number of participating peers: for the scenario with 512 cooperating peers, our combinational method would cause the moderate total network load of 250 Mb; point-to-point exchanging of complete training collections would require transfers of 2.5 Terabyte data. The bottom line of this simple calculation is that our combination methodology does not present any critical performance challenges for the decentralized P2P information systems and is viable from a scalability viewpoint.

7. EVALUATION

7.1 Testbed

We simulated a scenario of multiple users with personal data using established reference document collections as well as data gathered from social Web environments. We performed multiple series of experiments on the following data collections:

(1) The academic WebKB dataset [Craven et al. 1998] containing 8282 HTML Web pages from multiple universities, manually classified into the categories 'student', 'faculty', 'staff', 'department', 'course', 'project', and 'other'.

(2) Newsgroups collection at [20newsg]. This collection contains 17847 postings collected from 20 Usenet newsgroups. Particular topics ('rec.autos', 'sci.space', etc.) contain between 600 and 1000 documents.

(3) The Reuters articles [Lewis 1991]. This is the most widely used test collection for text categorization research. The collection contains 21578 Reuters newswire stories, subdivided into multiple categories ('earn', 'grain', 'trade', etc.).

(4) The Internet Movie Database (IMDB) at [imdb]. Documents of this collection are articles about movies that include the storyboard, cast overview, and user

comments. The collection contains 6853 movie descriptions subdivided into 20 topics according to particular genres ('drama', 'horror', etc.).

(5) Beyond static reference data collections, we used the social bookmarking portal del.icio.us [del.icio.us] for constructing a realistic use-case junk elimination scenario. For this purpose, we used a systematic large-scale del.icio.us crawl that covers relationships between 18,778,520 resources (bookmarks), 532,938 users, and 140,305,446 bookmark annotations (tag assignments).

In all presented experiments, we used the standard bag-of-words approach [Baeza-Yates and Ribeiro-Neto 1999] for document representation. For constructing term-based features, we applied stopword elimination and the Porter stemming algorithm [Porter 1997]. The resulting term frequencies (tf) were used as weights for constructing L1-normalized feature vectors without dimensionality reduction (i.e. without additional feature selection). As discussed in Section 7.2, for each data set we identified all topics with sufficiently many documents. These were 20 topics for Newsgroups, 7 for Reuters, 9 from IMDB, and 4 from WebKB. For del.icio.us, we considered a set of 11 top-level dmoz [dmoz] categories ('art', 'business', 'computer', 'games', 'health', 'home', 'news', 'science', 'shopping', 'society', 'sports') as topics of interest.

7.2 Collaborative Scenarios

7.2.1 *Experiments with Supervised Learning Methods (Classification).* Among the selected topics from our reference collections, we randomly chose 100 topic pairs from Newsgroups and all possible $\binom{n}{2}$ combinations for the other datasets (for the n topics containing a sufficient amount of documents for our experiments) , i.e. 66 topic pairs from IMDB, 15 for Reuters, 6 for WebKB. For each topic pair we randomly chose 200 training documents per class and kept - depending on the available topic sizes in particular collections - a distinct and also randomly chosen set of documents for the validation of the classifiers. For modeling peers that contain not fully disjoint training data, we added a redundant copy for 15% of the documents (the pair-wise overlap between two peers remains reasonably small). In each experiment, the resulting training dataset was distributed over 16 peers using equal-sized subsets. Among these peers we randomly chose 1,2,4,8, and all 16 peers to simulate various P2P classification scenarios. For each peer, we used binary linear support vector machine (SVM) classifiers discriminating the two classes for each topic pair to get the individual classification results.

The configuration with 1 peer corresponds to the 'local' classification that does not involve sharing of classifiers. Our quality measure is the fraction of correctly classified documents (accuracy) among the documents not dismissed by the restrictive algorithm. The *loss* is the fraction of dismissed documents. In these experiments, we used SVM distances as confidence values for the base classifiers; the sum of the confidence values was considered as the confidence value of the meta result. Subsequently, we induced the loss by dismissing the corresponding fraction of documents with lowest meta confidence values.

Finally, we computed micro-averaged results along with their 95% confidence intervals for all groups of topic pairs. Figure 2 shows the observed dependencies between the numbers of cooperating peers, the induced loss, and the resulting accu-

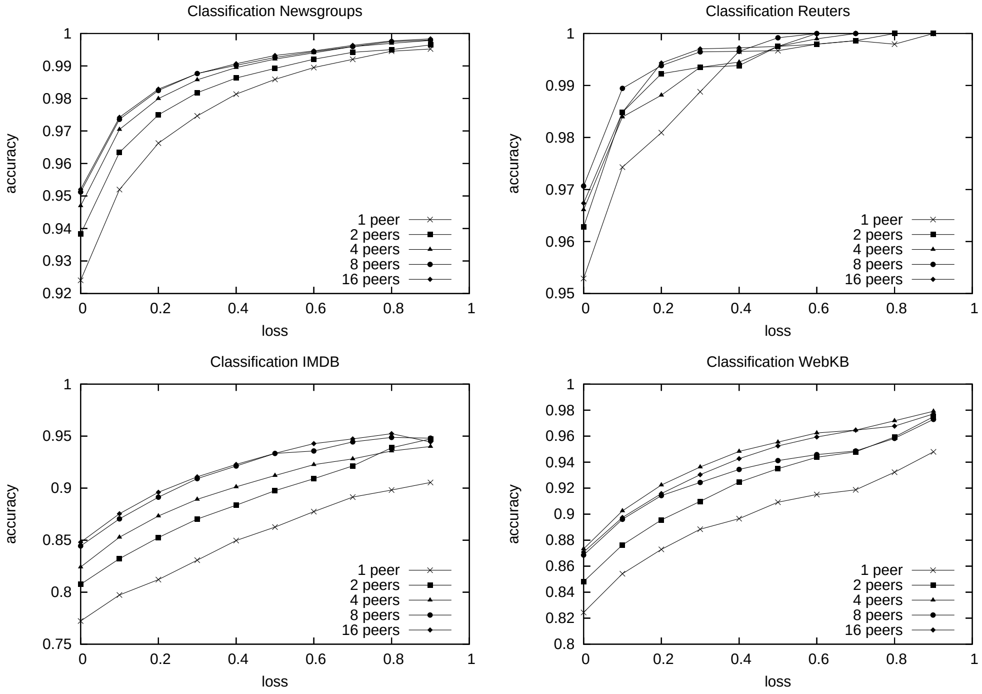

Fig. 2. Results of Restrictive Meta Classification

racy for various reference collections. It can be observed that the meta classification and restrictive meta classification by multiple cooperating peers clearly outperforms the single-peer solution for all settings of the user-defined *loss*, including the non-restrictive meta classification with $loss = 0$. The quality of the meta algorithm clearly increases with the number of participating peers.

7.2.2 *Experiments for Junk Filtering.* The same collections and topic pairs as in the classification experiments were also used to evaluate junk filtering. In contrast to classification experiments (with evaluation documents taken only from training classes) we additionally "spoiled" the validation set for each pair by increasing this set by 100% by adding randomly chosen "junk documents" from other - irrelevant - topics of the collection.

Our quality measures are defined as in Section 4.3. The *accuracy* is the fraction of correctly classified non-junk documents among all the validation documents not dismissed by the restrictive algorithm. The *loss* is the fraction of dismissed non-junk documents. The *junk reduction* is the fraction of dismissed junk documents. We considered these quality measures at different degrees of *document reduction* (0.0..0.9), induced analogously to the classification experiments in Section 7.2.1, and for different numbers of cooperating peers (1..16). Analogously to the classification scenario, the configuration with 1 peer corresponds to the 'local' junk elimination that does not involve classifier sharing.

Figure 3 shows the accuracy of restrictive meta methods with junk elimination

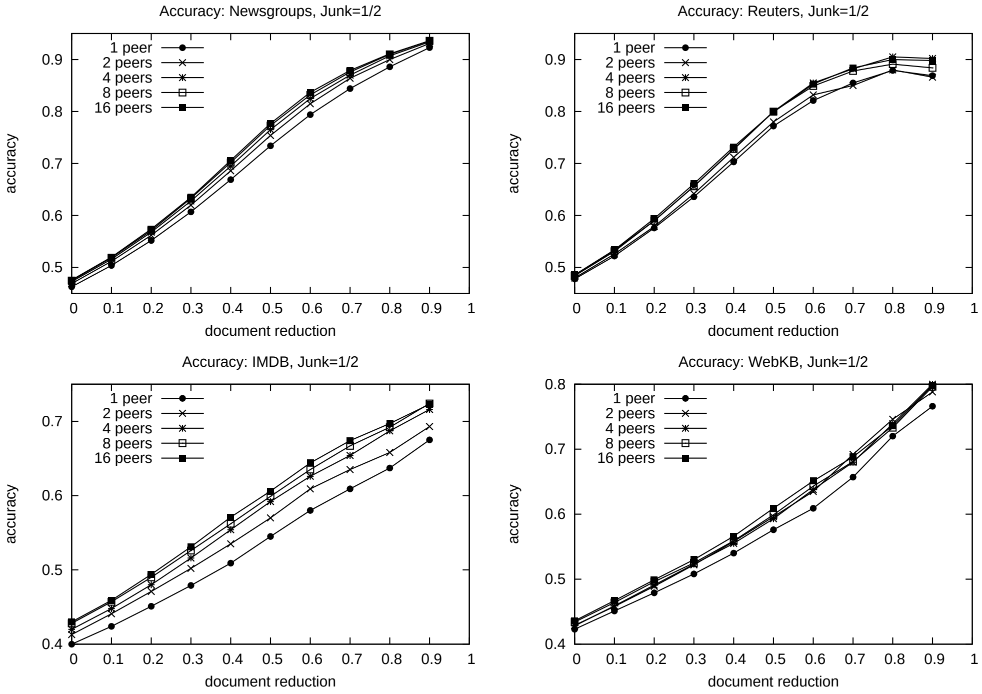

Fig. 3. Accuracy for Restrictive Junk Elimination

for different numbers of cooperating peers. Figure 4 compares the document loss and the junk reduction observed in these experimental series. For all datasets, we consistently obtain the following results:

—With increasing restrictivity (document reduction), the accuracy and the junk reduction increase at the cost of an increasing loss.

—For all levels of document reduction, the junk reduction shows a superlinear and the loss a sublinear behavior, i.e. the prevalent fraction of dismissed documents is junk.

—With increasing number of peers, we obtain a better tradeoff between accuracy and junk reduction on the one hand and document loss on the other hand, i.e. better accuracy, higher junk reduction, and lower loss for the same document reduction.

7.2.3 *Experiments with Unsupervised Learning Methods (Clustering).* The collections and topics from classification experiments were also used to evaluate distributed meta clustering. All documents from randomly combined selections of 3 or 5 topics were considered as unlabeled data and distributed among peers analogously to classification experiments from the previous section, with approximately 15% overlap. The goal of the clustering algorithm was to reproduce the partitioning into topics on each peer with possibly high accuracy. Our quality measure describes

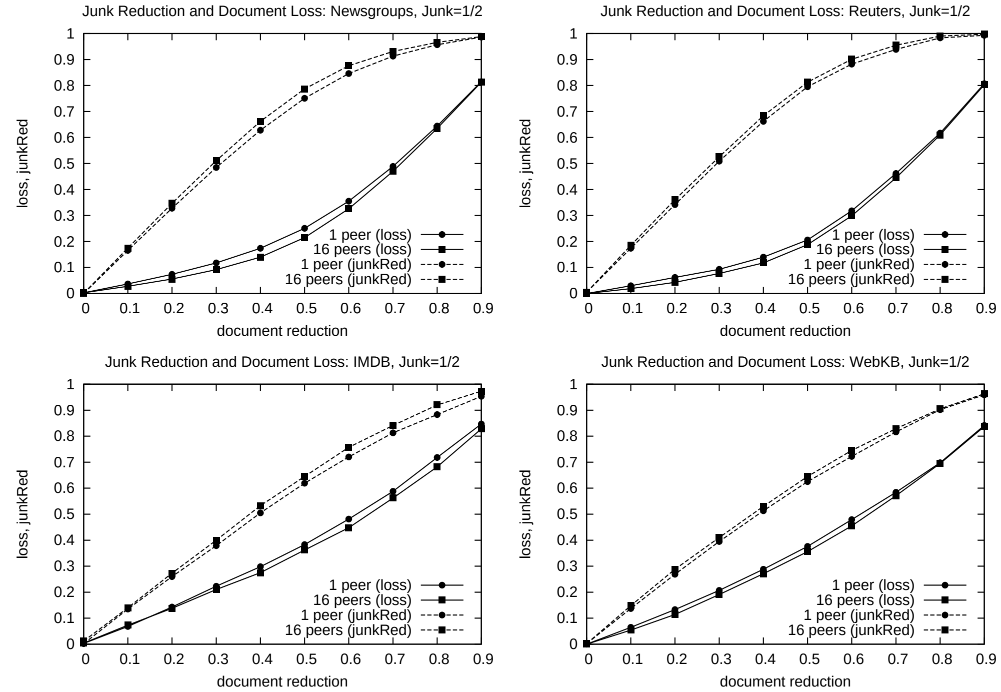

Fig. 4. Junk Reduction and Loss for Restrictive Junk Elimination

the correlation between the actual topics of our datasets and the clusters found by the algorithm. 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 cluster label j.

Unlike classification results, the clusters do not have explicit topic labels; We define the clustering accuracy as follows:

$$accuracy = \max_{(j_1,\dots,j_k)\in perm((1,\dots,k))} \frac{\sum_{i=1}^{k} N_{i,j_i}}{\sum_{i=1}^{k} N_i} \tag{39}$$

The *loss* is the fraction of documents dismissed by the restrictive algorithm.

For all peers, k-means was used as the underlying base method (considering the cosine similarities to the nearest centroids as confidence values). We compared the one-peer clustering (i.e. clustering that can be executed by one peer on its local dataset without cooperation with others) with meta clustering, exchanging centroids from cooperating peers and correlation-based mapping of the final clusters. We were considering the same number of clusters for each peer; we were using meta mapping as described in Section 5.2.1 in order to map between the cluster labels of the different clustering results. (Generalizing our mapping approach to scenarios with different numbers of clusters for distinct peers is possible in principle; however, this is out of the scope of this article.)

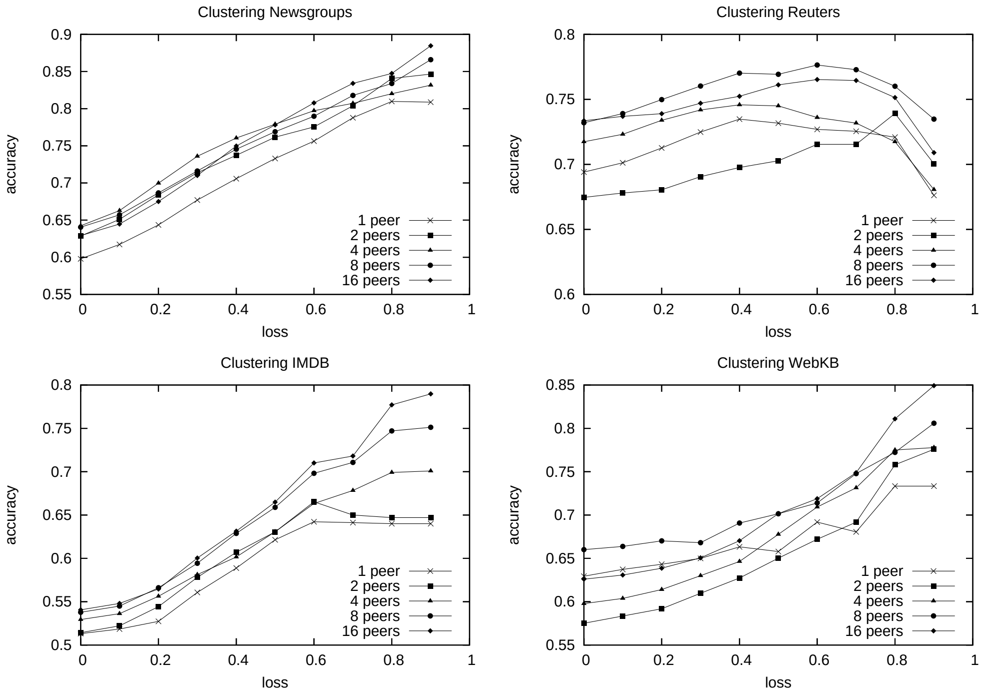

Fig. 5. Results of Restrictive Meta Clustering, k=3 Clusters

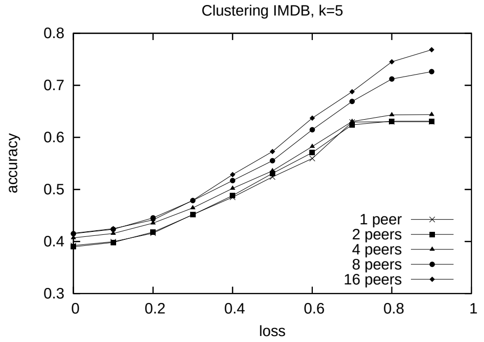

Fig. 6. Results of Restrictive Meta Clustering, k=5 Clusters

Analogously to classification experiments, we also considered restrictive meta clustering, dismissing the same number of documents with the worst clustering confidence on each peer.

The results are summarized in Figure 5. The main observations are similar to the ones discussed for the supervised case:

—The quality of the meta clustering results is consistently higher than for isolated one-peer solutions.

—The quality of the meta algorithm tends to increase with the number of participating peers.

In the experiments with the Reuters dataset, the accuracy decreases for high loss values (greater 0.7). Possibly this can be explained by the fact that the Reuters topics - unlike the other considered reference collections - are very different in size (e.g. the topics *earn* and *grain* contain about 3900 and 280 documents, respectively). The artifact observed for the Reuters data seems to be rather a shortcoming of the base machine learning method (i.e. the way k-means handles large differences in topic sizes in our case) than the meta learning as it already occurs for the one-peer solution. Meta clustering still leads to improvements but relies on the underlying base methods.

Figure 6 shows, as an example, the clustering of IMDB descriptions with k = 5 clusters. We notice that the qualitative behavior observed when changing restrictivity and number of peers is similar as for the case of a smaller number of clusters described above.

In our clustering experiments we have made the simplifying assumptions that we know the number of clusters and that the peers contain documents from the same categories. Due of the absence of training data and information about the underlying category structure, clustering generally poses a harder problem than classification, and parameter tuning as well as evaluation become more critical issues. Cluster correlations used for meta mapping described in Section 5 could act as an indicator for clusters that should or should not be mapped and there exists work on deciding the number k of clusters for a given data set [Li et al. 2004]. These issues are out of the scope of the current article but relevant topics for future work on automatic organization of personal data.

7.3 Experiments with Social Bookmarking Data from Del.icio.us

For the del.icio.us data set, we identified users with at least 50 (and, to avoid spammers, less than 200) annotated bookmarks per topic, considering the top-level dmoz [dmoz] categories (‘art’, ‘business’, ‘computer’, ‘games’, ‘health’, ‘home’, ‘news’, ‘science’, ‘shopping’, ‘society’, ‘sports’) as topics of interest. For each topic, we considered all tags containing the category name as prefix (i.e. tags like ’healthy-skin’, ’healthyheart’, or ’health—fitness’ for the category ’health’). For evaluation purposes we removed documents with multiple topic assignments. We obtained 1,918 users with sufficient training data for at least two of the mentioned topics of interest. After leaving out infrequent topic combinations, we obtained 42 topic pairs with at least 16 users, which were considered as ’peers’ in this experiment. The resulting bookmark collection (totally 280,465 links) was completely crawled. The downloaded Web documents form personal training collections of the users/peers;

we trained the decision model of each peer on all documents of the corresponding user. In each experiment, the documents from the other users were used for constructing validation sets (1000 randomly chosen documents for each topic in the pair) and we added junk documents to the validation sets (2000 randomly chosen documents from irrelevant topics). We ensured the disjointness between training and test collections. Our objective was to evaluate the ability of meta methods for collaborative, automatic, content-based categorization or tagging of new postings.

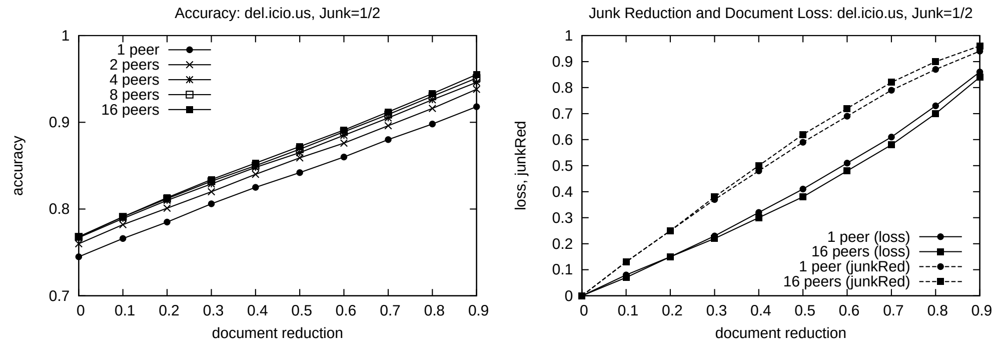

Fig. 7. Accuracy and junk reduction for del.icio.us users

Figure 7 shows the accuracy of restrictive meta methods with junk elimination for different numbers of cooperating peers (1-16) and compares the document loss and the junk reduction for different settings in this experiment. Our results show the same qualitative behavior as previous experiments on the standard data sets and thus demonstrate the viability of our approach.

In contrast to more traditional reference datasets, topic labels in del.icio.us are provided by the individual users, and, thus have a more personal character. Our del.icio.us sample indicates that folksonomy users agree on frequent topic labels, and the assigned tags are often surprisingly consistent with labels used in taxonomies such as dmoz. Our results for the del.icio.us data set show the same qualitative behavior as the experiments on the standard data set, and, thus, demonstrate the viability of our approach for the natural user-per-peer distribution of the data which more closely resembles the application of our methods for PIM scenarios.

8. CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this article we proposed the methodology of restrictive methods and meta methods for collaborative structuring of personal document collections. The key objective is to combine decisions of multiple models from distinct users into a meta result. High quality is achieved by carefully restricting the outcome to the best results with highest confidence degrees. Furthermore, we introduced a formal model that explains the ability of meta methods for accuracy improvement and junk elimination. The model can be used for constructing estimators for predictable exploitation of accuracy/loss tradeoffs. We demonstrated the application of meta methods for a number of problems: supervised and unsupervised partitioning of document collections, junk elimination, and data filtering.

We considered the paradigm of meta methods in the context of decentralized collaborative environments, such as peer-to-peer information systems. Our methodology of decentralized information organization does not require the comprehensive exchange of private data collections between peers and thus provides substantial advantages for aspects of privacy, network bandwidth, storage, and computational expense. Furthermore, in terms of accuracy our restrictive meta methods clearly outperform models that can be separately built on training sources of isolated peers and - more importantly - also the restrictive variant of such one-peer solutions with same document reduction.

8.2 Future Work

Our long-term goal is to develop robust and accurate algorithms for quality-oriented personal information management.

In a sharing system, different users might organize their documents into distinct topic taxonomies. In this case we aim to solve the ‘topic mapping’ problem in the presence of multiple label sets, which would enable us to build meta classification models. This goal can be achieved by estimating the similarity between peer document collections or the similarity of classifiers built locally on distinct peers and validated on collections of other peers. As a side effect, we could generate richer and more fine-grained taxonomies from multiple smaller taxonomies of many peers, and obtain a grouping of peers into “cliques” sharing the same interests.

There is also a need for space-efficient encodings of machine learning models and efficient and scalable algorithms for the propagation of these models between users (using e.g. Epidemic Protocols or P2P architectures such as CHORD). Here the dynamic aspects of the P2P environment, allowing users to get connected or disconnected in the network, must be taken into account, too.

In a large P2P environment there may also be adversarial users that aim to pollute the statistical learning models by intentionally introducing incorrectly labeled data. The automatic detection of these peers on the one hand, and, on the other hand, the automatic recognition of “networks of trust” that do not pursue contradictory interests are important. A key element of this approach could be the mutual evaluation of peers in the environment and the mapping of the results into a graph-based representation. A similar approach could be applied for detecting qualitative differences of the data among the peers, that could allow us the construction of enhanced weighting schemes for meta models.

For personal information management, alternative classification tasks might be interesting. For instance, label ranking [Brinker and Hüllermeier 2007; 2006] tackles the interesting problem of assigning ranked orders of labels (instead of single class labels) to test samples. Assigning multiple labels to objects (tagging) is common for Web 2.0 content sharing applications. Open research issues include the combination of labeled rankings via classifier exchange and meta methods, restrictivity and tuning, and application in unsupervised data organization.

For the use of our methodologies in real systems, a variety of additional practical development issues and orthogonal theoretical points require careful attention. Interesting research topics include dealing with sparse and imbalanced training data, tag and taxonomy mapping problems, spamming, trust mechanisms for the exchanged data models, restricting model access (i.e. model destiny), and mechanisms to allow users to identify the part of their data where automatic organization makes sense. Beyond this, to better capture the user satisfaction in a realistic setting, future evaluation methodologies should also consider integrated applications (e.g. software tools for private PCs that provide the user with different alternative views on personal data).

Acknowledgements

This work has been partially supported by the European project WeKnowIt (”Emerging, Collective Intelligence for Personal, Organizational and Social Use”, FP7-215453) and the EU-funded project Memoir. Susan Dumais and the anonymous reviewers gave helpful comments on earlier revisions of this article.

REFERENCES

20newsg. The 20 Newsgroups Data Set. *http://www.ai.mit.edu/ jrennie/20Newsgroups/*.

Allwein, E. L., Schapire, R. E., and Singer, Y. 2001. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research 1*, 113–141.

Baeza-Yates, R. and Ribeiro-Neto, B. 1999. *Modern Information Retrieval.* Addison Wesley.

Bender, M., Michel, S., Triantafillou, P., Weikum, G., and Zimmer, C. 2005. Improving collection selection with overlap awareness in P2P search engines. *28th Annual International ACM SIGIR Conference, Salvador, Brazil*, 67–74.

Bender, M., Michel, S., Triantafillou, P., Weikum, G., and Zimmer, C. 2006. P2P Content Search: Give the Web Back to the People. *5th International Workshop on Peer-to-Peer Systems (IPTPS), Santa Barbara, USA*.

Bender, M., Michel, S., Weikum, G., and Zimmer, C. 2004. Bookmark-driven query routing in peer-to-peer Web search. *SIGIR Workshop on P2P Information Retrieval*.

Blok, H., Hiemstra, D., Choenni, S., de Jong, F., Blanken, H., and Apers, P. 2001. Predicting the Cost-Quality Tradeoff for Information Retrieval Queries: Facilitating Database Design and Query Optimization. *10th International Conference on Information and Knowledge Management (CIKM), Atlanta, USA*, 207–214.

Bloom, B. 1970. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM 13,* 7, 422–426.

Boardman, R. and Sasse, M. A. 2004. ”stuff goes into the computer and doesn't come out”: a cross-tool study of personal information management. In *CHI '04: Proceedings of the SIGCHI conference on Human Factors in computing systems*. Vienna, Austria.

Brank, J., Grobelnik, M., Milic-Frayling, N., and Mladenic, D. 2003. Training text classifiers with SVM on very few positive examples. *Technical Report MSR-TR-2003-34, Microsoft Corp.*.

Breiman, L. 1996. Bagging Predictors. *Machine Learning 24,* 2, 123–140.

Brinker, K. and Hüllermeier, E. 2006. Case-based label ranking. In *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Lecture Notes in Computer Science. Springer, 566–573.

Brinker, K. and Hüllermeier, E. 2007. Label ranking in case-based reasoning. In *Case-Based Reasoning Research and Development, 7th International Conference on Case-Based Reasoning, ICCBR 2007, Belfast, Northern Ireland, UK, August 13-16*. Lecture Notes in Computer Science. Springer, 77–91.

Brutlag, J. D. and Meek, C. 2000. Challenges of the email domain for text classification. In *Seventeenth International Conference on Machine Learning (ICML 2000)*. 103–110.

Buckland, M. K. 1992. Emmanuel goldberg, electronic document retrieval, and vannevar bush's memex. *JASIS 43,* 4, 284–294.

Burges, C. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery 2,* 2, 121–167.

Bush, V. 1945. As We May Think. *The Atlantic Monthly 176,* 1, 101 – 108.

Chakrabarti, S. 2002. *Mining the Web: Discovering Knowledge from Hypertext Data.* Morgan Kaufman.

Chan, P. 1996. An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning. *PhD thesis, Department of Computer Science, Columbia University, New York*.

Choi, N., Song, I.-Y., and Han, H. 2006. A survey on ontology mapping. *SIGMOD Rec. 35,* 3, 34–41.

Cormack, G. V. 2006. Trec 2006 spam evaluation track overview. *Fifteenth Text REtrieval Conference (TREC-2006)*.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. 1998. Learning to Extract Symbolic Knowledge from the World Wide Web. *15th National Conference on Artificial Intelligence (AAAI), Madison, USA*, 509–516.

Cronen-Townsend, S., Zhou, Y., and Croft, W. 2002. Predicting Query Performance. *25th International ACM Conference on Research and Development in Information Retrieval (SIGIR), Tampere, Finland*, 299–306.

Cutrell, E., Robbins, D., Dumais, S., and Sarin, R. 2006. Fast, flexible filtering with phlat. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems.* ACM, New York, NY, USA, 261–270.

Davé, R. N. 1991. Characterization and detection of noise in clustering. *Pattern Recognition Letters 12,* 11, 657–664.

del.icio.us. Del.icio.us: a Social Bookmarking System. *http://del.icio.us*.

Dhillon, I. and Modha, D. 2000. A Data-Clustering Algorithm on Distributed Memory Multiprocessors. *Large-Scale Parallel Data Mining, Lecture Notes in Artificial Intelligence*, 245–260.

Dimitriadou, E., Weingessel, A., and Hornik, K. 2002. A combination scheme for fuzzy clustering. In *AFSS '02: Proceedings of the 2002 AFSS International Conference on Fuzzy Systems.* Springer-Verlag, Calcutta, 332–338.

dmoz. dmoz - open directory project. *http://dmoz.org/*.

Dong, X. and Halevy, A. 2005. A platform for personal information management and integration. *2nd Conference on Innovative Systems Research (CIDR), Asilomar, USA*, 119–130.

Dragunov, A. N., Dietterich, T. G., Johnsrude, K., McLaughlin, M., Li, L., and Herlocker, J. L. 2005. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces.* ACM Press, New York, NY, USA, 75–82.

Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., and Robbins, D. C. 2003. Stuff i've seen: a system for personal information retrieval and re-use. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval.* ACM Press, New York, NY, USA, 72–79.

Dumais, S. T., Platt, J. C., Hecherman, D., and Sahami, M. 1998. Inductive learning algorithms and representations for text categorization. In *ACM CIKM International Conference on Information and Knowledge Management.* 148–155.

Ester, M., Kriegel, H.-P., and Sander, J. 2001. *Knowledge Discovery in Databases.* Springer.

Fern, X. Z. and Brodley, C. E. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning.* ACM Press, Banff, Alberta, Canada, 36.

Fred, A. L. N. and Jain, A. K. 2002. Data clustering using evidence accumulation. In *International Conference on Pattern Recognition ICPR (4).* IEEE Computer Society, Quebec, Canada, 276–280.

Freund, Y. 1999. An Adaptive Version of the Boost by Majority Algorithm. *Workshop on Computational Learning Theory (COLT), Santa Cruz, USA*, 102–113.

Fumera, G., Pillai, I., and Roli, F. 2003. Classification with reject option in text categorisation systems. In *ICIAP '03: Proceedings of the 12th International Conference on Image Analysis and Processing*. IEEE Computer Society, Washington, DC, USA, 582.

Gemmell, J., Bell, G., and Lueder, R. 2006. Mylifebits: a personal database for everything. *Commun. ACM 49,* 1, 88–95.

Gemmell, J., Bell, G., Lueder, R., Drucker, S. M., and Wong, C. 2002. Mylifebits: fulfilling the memex vision. In *ACM Multimedia*. 235–238.

Goerlitz, O., Sizov, S., and Staab, S. 2008. PINTS: Peer-to-Peer Infrastructure for Tagging Systems. *7th International Workshop on Peer-to-Peer Systems (IPTPS), Tampa Bay, USA*.

google. Google desktop. *http://desktop.google.com/*.

Groza, T., Handschuh, S., Moeller, K., Grimnes, G., Sauermann, L., Minack, E., Mesnage, C., Jazayeri, M., Reif, G., and Gudjonsdottir, R. 2007. The NEPOMUK Project - On the way to the Social Semantic Desktop. *International Conference on Semantic Technologies (I-Semantics), Graz, Austria*, 201–211.

Haag, S., Cummings, M., and McCubbrey, D. J. 2002. *Management information systems for the information age*, 3 ed. Irwin McGraw-Hill.

Han, J. and Kamber, M. 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Hartigan, J. and Wong, M. 1979. A k-Means clustering algorithm. *Applied Statistics, 28:100-108*.

imdb. Internet movie database. *http://www.imdb.com*.

Joachims, T. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. *ECML*.

Kargupta, H., Huang, W., Sivakumar, K., and Johnson, E. 2001. Distributed Clustering Using Collective Principal Component Analysis. *Knowledge and Information Systems 3,* 4, 422–448.

Klimt, B. and Yang, Y. 2004. The enron corpus: A new dataset for email classification research. In *ECML 2004, 15th European Conference on Machine Learning*. Lecture Notes in Computer Science. Springer, 217–226.

Kogan, S. L. and Muller, M. J. 2006. Ethnographic study of collaborative knowledge work. *IBM Systems Journal 45,* 4, 759.

Kuhn, H. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, 83–97.

Kuncheva, L. I. 2004. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience.

Landgrebe, T. C. W., Tax, D. M. J., Paclík, P., and Duin, R. P. W. 2006. The interaction between classification and reject performance for distance-based reject-option classifiers. *Pattern Recogn. Lett. 27,* 8, 908–917.

Lewis, D. 1991. Evaluating Text Categorization. *Proceedings of Speech and Natural Language Workshop, Pacific Grove, USA*, 312–318.

Li, T., Ma, S., and Ogihara, M. 2004. Document clustering via adaptive subspace iteration. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 218–225.

Li, T., Zhu, S., and Ogihara, M. 2003. Algorithms for Clustering High Dimensional and Distributed Data. *Intelligent Data Analysis Journal 7,* 4.

Littlestone, N. and Warmuth, M. 1989. The weighted majority algorithm. *FOCS*.

Madison, W., Yang, Y., and Pedersen, J. 1997. A comparative study on feature selection in text categorization. In *ICML*.

Manning, C. and Schuetze, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Masulli, F. and Valentini, G. 2000. Comparing decomposition methods for classification. In *International Conference on Knowledge-Based Intelligent Engineering Systems and Applied Technologies, KES*. Brighton, UK, 788–792.

Merugu, S. and Ghosh, J. 2003. Privacy-Preserving Distributed Clustering using Generative Models. *International Conference on Data Mining (ICDM), Melbourne, USA*, 211–218.

Millen, D., Yeng, M., Whittaker, S., and Feinberg, J. 2007. Social bookmarking and exploratory search. In *European Conference on Computer Supported Co-operative Work 2007. (in press).*

Millen, D. R., Feinberg, J., and Kerr, B. 2006. Dogear: Social bookmarking in the enterprise. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems.* ACM Press, New York, NY, USA, 111–120.

Pierskalla, W. 1968. The multi-dimensional assignment problem. *Operations Research 16*, 422–431.

Platt, J. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in Large Margin Classifiers, MIT Press*, 61–74.

Porter, M. 1997. An Algorithm for Suffix Stripping. *ACM Readings in Information Retrieval*, 313–316.

Quan, D., Huynh, D., and Karger, D. 2003. Haystack: A platform for authoring end user semantic web applications. *International Semantic Web Conference*, 738–753.

Rivest, R. 1992. RFC 1321: The MD5 Message Digest Algorithm. *http://www.ietf.org/rfc/rfc1321.txt*.

Schein, A., Popescul, A., Ungar, L., and Pennock, D. 2002. Methods and metrics for cold-start recommendations. *ACM Conference on Research and Development in Information Retreival*.

Segal, R. and Kephart, J. O. 1999. Mailcat: An intelligent assistant for organizing e-mail. In *International Conference on Autonomous Agents.* 276–282.

Shvaiko, P. and Euzenat, J. 2005. A survey of schema-based matching approaches. 146–171.

Siersdorfer, S. and Sizov, S. 2003. Construction of Feature Spaces and Meta Methods for Classification of Web Documents. *10th Conference Datenbanksysteme fuer Business, Technologie und Web (BTW), Leipzig, Germany*, 197–206.

Siersdorfer, S. and Sizov, S. 2004. Restrictive Clustering and Meta clustering for Self-Organizing Document Collections. *27th International Conference on Research and Development in Information Retrieval (SIGIR), Sheffield, UK*, 226–233.

Siersdorfer, S. and Sizov, S. 2006. Automatic document organization in a p2p environment. In *Advances in Information Retrieval, 28th European Conference on IR Research, ECIR.* 265–276.

Siersdorfer, S. and Sizov, S. 2007. Restrictive methods and meta methods for thematically focused web search. *Handbook of Research on Web Information Systems Quality, Idea Group*.

Siersdorfer, S., Sizov, S., and Weikum, G. 2004. Goal-oriented methods and meta methods for document classification and their parameter tuning. In *CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management.* ACM Press, Washington, D.C., USA, 59–68.

Siersdorfer, S. and Weikum, G. 2005. Using restrictive classification and meta classification for junk elimination. In *Proceedings of the 27th European Conference on Information Retrieval (ECIR '05)*, D. Losada and J. M. F. Luna, Eds. Lecture Notes in Computer Science, vol. 3408. Information Retrieval Specialist Group of the British Computer Society (BCS-IRSG), Springer, Santiago de Compostela, Spain, 287–299.

Stoica, I., Morris, R., Karger, D. R., Kaashoek, M. F., and Balakrishnan, H. 2001. Chord: A scalable peer-to-peer lookup protocol for internet applications. In *Proc. of the ACM SIGCOMM.* San Diego, 149–160.

Strehl, A. and Gosh, J. 2002. Cluster Ensembles - a Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research 3*, 583–617.

Surendran, A. C., Platt, J. C., and Renshaw, E. 2005. Automatic discovery of personal topics to organize email. In *CEAS 2005 - Second Conference on Email and Anti-Spam.*

Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R. 2004. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *CHI '04: Proceedings of the SIGCHI conference on Human Factors in computing systems.* Vienna, Austria.

Topchy, A., Jain, A. K., and Punch, W. 2003. Combining multiple weak clusterings. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining.* IEEE Computer Society, Melbourne, Florida, USA, 331.

Vaidya, J. and Clifton, C. 2004. Privacy Preserving Naive Bayes Classifier for Vertically Partitioned Data. *SIAM International Conference on Data Mining, Orlando, USA*.

Vailaya, A. and Jain, A. K. 2000. Reject option for vq-based bayesian classification. In *International Conference on Pattern Recognition (ICPR'00).* 2048–2051.

Van Rijsbergen, C. 1977. A Theoretical Basis for the Use of Co-occurence Data in Information Retrieval. *Journal of Documentation 33,* 2, 106–119.

Wang, G. and Lochovsky, F. 2004. Feature Selection with Conditional Mutual Information MaxiMin in Text Categorization. *13th ACM Conference on Information and Knowledge Management (CIKM), Washington D.C., USA*, 342–349.

Wang, H., Fan, W., Yu, P., and Han, J. 2003. Mining Concept-Drifting Data Streams using Ensemble Classifiers. *9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Washington, USA*, 226–235.

Wolpert, D. 1992. Stacked Generalization. *Neural Networks 5*, 241–259.

Yu, H., Chang, K., and Han, J. 2002. Heterogeneous Learner for Web Page Classification. *IEEE International Conference on Data Mining (ICDM), Maebashi, Japan*, 538–545.

Zhang, R. and Metaxas, D. 2006. Ro-svm: Support vector machine with reject option for image categorization. III:1209.