Performance Thresholding in Practical Text Classification

Hinrich Schütze
Institute for NLP
Universität Stuttgart
Germany
hs999@ifnlp.org

Emre Velipasaoglu
Yahoo! Inc.
Sunnyvale, CA 94089
USA
emrev@yahoo-inc.com

Jan O. Pedersen
Yahoo! Inc.
Sunnyvale, CA 94089
USA
jpederse@yahoo-inc.com

ABSTRACT

In practical classification, there is often a mix of learnable and unlearnable classes and only a classifier above a minimum performance threshold can be deployed. This problem is exacerbated if the training set is created by active learning. The bias of actively learned training sets makes it hard to determine whether a class has been learned. We give evidence that there is no general and efficient method for reducing the bias and correctly identifying classes that have been learned. However, we characterize a number of scenarios where active learning can succeed despite these difficulties.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

practical text classification, active learning, accuracy estimation, learnability

1. INTRODUCTION

Text classification is the problem of devising an automatic method for determining membership of a document in a predefined text category or class [23]. For example, a spam filter recognizes email messages as being part of the category spam and directs them to a special folder. The behavior of many other document-centric applications depends on category membership in a similar way. Text classification has therefore become one of the key technologies for processing and managing electronic documents.

Text classifiers are commonly created by estimating the parameters of a statistical classification model on a labeled training set. Most academic research on text classification assumes the prior existence of a sufficiently large training set. However, in most practical settings there initially exists no labeled training set. Creating the labeled set that is needed for parameter estimation is therefore an integral part of the text classification problem.

In fact, training set creation is often the main cost in text classification since it has to be done manually. Sometimes labelers are hired and paid for the sole purpose of training set creation. In other cases, one needs to find a subject expert within an organization, typically an employee who has other responsibilities and will be reluctant to spend hours on end with a repetitive labeling task.

Another characteristic of practical text classification is that classifiers must operate at a minimum level of performance. For most business problems it is unacceptable to deploy a classifier with unknown performance. We therefore need a way to assess the performance of a classifier in order to make sure that only well performing classifiers are deployed.

We call this setting *practical text classification* to distinguish it from work on text classification that assumes the prior existence of a large labeled training set. Practical text classification is characterized by the following conditions:

- Initially, there exists no labeled training set.
- Labeling is expensive and the cost of labeling should be minimized.
- Classifiers can only be deployed if their performance (as measured by some evaluation measure such as F_1) is better than a predetermined threshold.
- Most classes are "small": their population rate is 0.01 or even smaller.

For small classes, one cannot efficiently create a training set by drawing a random sample. A category with a relative frequency of 0.01 is expected to have just 10 positive examples in a random sample of 1000. In general, one needs many more than 10 positive examples to train a reliable text classifier.

A more efficient method of training set creation is active learning (AL) [14]. AL samples documents from the subspace that contains "informative" or "uncertain" documents, i.e., those documents that the current classifier is unsure about and that are likely to provide the most information per labeling decision. An iteration of AL selects one or more such informative documents from the *pool* of unlabeled documents, labels them, adds them to the training

CIKM'06, November 5–11, 2006, Arlington, Virginia, USA.
Copyright 2006 ACM 1-59593-433-2/06/0011 ...$5.00.

set and retrains the classifier, thus redefining the region of uncertainty in each step. To get AL started, one needs a small seed set of labeled documents, which may come from a keyword search or some other source. AL implicitly assumes that a large pool of unlabeled documents is available and that querying this pool is cheap compared to labeling since the latter takes a human expert's time.

It has been shown that AL is effective in creating high performance classifiers with a relatively small number of labeling decisions [4, 14, 27]. However, given the additional need in practical text classification to only field classifiers that exceed some predetermined threshold, two problems arise: learnability and decidability.

Learnability refers to the fact that some classes are not learnable. We define a class to be learnable if there exists a learning procedure that produces a classifier exceeding a performance threshold θ with a certain level of confidence, where threshold and confidence level depend on the application (a θ of F=70% to 95% and a confidence level of 95% to 99% are typical). A class may not be learnable in principle because of a high Bayes error rate intrinsic to the learning problem or it may not be learnable because the selected document representation formalism is not powerful enough (e.g., a bag of words model). Note that this concept of learnability is different than PAC learnability in machine learning [18, 9]. In PAC learnability, a problem is considered learnable if, with high probability, the learner finds a hypothesis within a delta of the minimum possible error. In contrast, we want the learner to find a hypothesis within delta of an absolute level of performance (and to detect if that is not possible). So learnability as defined here is with respect to an *absolute* level of performance whereas it is *relative* to the optimal possible performance in PAC learnability.

Some corpora used in text classification research only contain learnable classes. But in practice classes are selected and defined according to their utility for a business problem. Our experience at several companies is that very often a subset of these classes turn out to be unlearnable. In some cases the reason is that even humans cannot make the classification decision; in other cases the representations typically used in text classification omit important information (e.g., the order of words in the bag-of-words model). In both cases, the Bayes error rate of the classification problem, as defined in the particular application, is too high to predict class membership reliably (i.e., with performance exceeding the chosen threshold). If AL is applied to such a category, the training set will not contain sufficient information to learn the category. Most previous work on AL assumes that all categories are learnable, so the learnability problem was not addressed. In those cases where researchers work with a mix of learnable and unlearnable classes, their goal is to optimize classification performance even if optimal performance is at a low level for a subset of classes [14, 15] or to maximize micro-average [30], which is mostly determined by performance on large categories. The problem of learnability is ignored in both approaches.

Decidability refers to the meta-problem of deciding whether a category has been learned at a given point in AL or not. We need an automatic decision procedure that tells us whether learning was successful. The problem is that assessing performance objectively is hard in the absence of a randomly sampled training set. The actively learned training set is highly biased, so that standard methods (e.g., leave-one-out) do not produce usable estimates as we will show below. Decidability is critical in practical text classification as we define it since a classifier can only be deployed if it meets minimum performance requirements.

The bulk of this paper is concerned with developing a method that addresses learnability and decidability based on an idea due to [12] for estimating the performance of a classifier from unlabeled data. We propose several variants of this method and evaluate them experimentally. Our conclusion is that there is no general solution to learnability and decidability in practical text classification that is more efficient than random sampling. In particular, we conclude that in reality AL does not reduce the effort required to produce a deployable classifier unless additional assumptions are made. While our study is empirical, we believe that the experimental evidence presented here provides strong support for our conclusions.

In addition to recasting the problem of practical text classification in these terms, our main contribution is to bound the applicability of AL. We characterize a number of scenarios where AL is an effective approach to practical text classification in spite of the difficulties discussed.

The paper is organized as follows. Two examples illustrating the problems of learnability and decidability are presented in Section 2. Section 3 describes a decision procedure based on Lewis' estimator of F. Five methods that attempt to produce unbiased estimates for this procedure are proposed and evaluated in Section 4. Section 5 analyzes the results of the experiments. Sections 6 and 7 discuss related work and learnability and decidability in practical text classification in light of our experiments.

2. THE MISSED CLUSTER EFFECT

For illustration, we present two examples, one synthetic and one from the experiments described below. In the synthetic example, we want to learn a binary classification function f defined on the real numbers. Let the hypothesis space be the set of sets of real non-overlapping intervals: $I_c = \langle [l_1, r_1], [l_2, r_2], \ldots, [l_{n_c}, r_{n_c}] \rangle$ with $l_i < r_i$ and $r_i < l_{i+1}$ for $i \leq n_c - 1$. A point is in a category iff it is in one of its intervals. Consider the set of categories c_i: $I_{c_0} = \langle [-2, -1] \rangle$, $I_{c_i} = \langle [-2, -1], [2i, 2i+1] \rangle, i > 0$. If, by bad luck, we end up with a seed set of points none of which is in an interval $[2i, 2i+1], i > 0$, then AL will focus on the interval $[-2, -1]$ and its surroundings. There simply is no information that we could exploit systematically that would distinguish cases where we learn c_0 from cases where we learn one of the other c_i. AL strategies will learn the decision boundaries -1 and -2 and thus reach precision close to 100%. But there is no general procedure for determining whether there are unexplored parts of the space that contain positive examples. We call such unexplored regions *missed clusters*. Missed clusters can only be found with random sampling, which by definition is not AL. As a result, AL cannot learn the categories $c_i, i > 0$ for a seed set that does not contain points from the interval $[2i, 2i+1]$. This is the learnability problem: a category $c_i, i > 0$ will only be learned for certain fortuitous seed sets. If (parts of) a category are not learned, then this affects accuracy estimation, in particular recall estimation. Recall will be overestimated in the case of an unknown missed cluster. For that reason a correct decision as to whether a classifier has reached a certain level of accuracy cannot be made. This is the decidability problem.

To provide an example from our experiments, we analyze an AL model for the category "Australia" in RCV1, a corpus of newswire articles from 1996 and 1997 covering topics like politics, business and sports (see Section 4 for details on our experimental setup). A set of false negatives from unlabeled data was clustered into 10 clusters using k-means. We computed the distance of these clusters to the "labeled cluster," the cluster consisting of the entire training set labeled in AL. We define the distance of two clusters as the smallest Euclidean distance of any pair of members. Cluster 7 (with 53 documents) was the cluster with the largest distance (1.64) to the labeled cluster. Note that the maximum distance of two normalized vectors is 2.0, so the two clusters are at close to maximum distance from each other. Of terms that did not occur in the labeled set, the following 5 occurred in most documents in Cluster 7: cbot (occurred in 48 documents), nymex (48), roundup (48), comex (47), and bushel (46). The cluster turned out to be a cluster of documents about the topic "Australian commodities roundup". Except for the single word "Australian," there is nothing else that makes Cluster 7 similar to the concept of relevance that the AL process has learned. "Australian" also occurs in many non-relevant documents, so there is no simple rule that would distinguish the false negatives in Cluster 7 from true negatives. Cluster 7 corresponds to the intervals $[2i, 2i+1]$ in the synthetic data: AL is not able to identify the members of Cluster 7 as informative or uncertain and therefore worthy of being included in the training set – Cluster 7 is too far from the decision boundary. And unless we are willing to give up AL for random sampling, there is no search algorithm for locating such clusters far from the decision boundary. As a consequence there is no obvious criterion for stopping AL – we never know whether there is an undiscovered cluster remaining or not.[1]

These two examples show that missed clusters, if they occur, pose serious problems for the decision problem of interest. However, it is possible that missed clusters rarely occur in practical classification problems. In the next section, we test the performance of an accuracy estimation procedure that is expected to perform well in the absence of missed clusters and give evidence that it fails because of the missed cluster effect.

3. ACCURACY ESTIMATION FOR AL

One solution to decidability is to define a level of acceptable performance θ, say F=80%, and stop AL when this level has been reached. This procedure is motivated by our experience with text classification in practical applications. Usually a minimum quality is required for a classifier to be deployed. An alternative goal would be to achieve optimal performance, but this is insufficient – a performance of 5% can be optimal for a particular category. A classifier with such a low performance cannot be deployed.

The key question is: how do we know whether the classifier is below or above θ? We need an absolute (as opposed to relative) assessment of accuracy of the classifier. We do not have the luxury of a random held-out set in AL – if there were sufficiently many labeled examples for such a set, then there would be no need for AL. We demonstrate below that leave-one-out estimation on the labeled sample has a large bias.

The decision procedure we evaluate as an alternative is based on Lewis' estimator of F for an unlabeled random sample. This estimator takes advantage of the fact that we can compute expected error rates for a random sample if our classifier is probabilistic. For example, if the class probability of a document according to the classifier is 80%, then there is a 20% error probability for a discrete assignment. The key advantage of this decision procedure is that we use an *unlabeled* instead of a *labeled* random sample to estimate the accuracy of the classifier.

We use the F_1 measure (based on [28], henceforth F), the harmonic mean of precision (P) and recall (R), for evaluation. Actively learned classifiers sometimes fail catastrophically with respect to precision or recall. In those cases, F is close to the minimum of the two, which is a good characterization of classifiers with either precision or recall close to 0.

F can be defined as a function of true positives (tp), false positives (fp) and false negatives (fn):

$$\begin{aligned} F &= (\frac{1}{2}\cdot(\frac{1}{P}+\frac{1}{R}))^{-1} = (\frac{1}{2}\cdot(\frac{\text{tp}+\text{fp}}{\text{tp}}+\frac{\text{tp}+\text{fn}}{\text{tp}}))^{-1} \\ &= \frac{2\text{tp}}{2\text{tp}+\text{fp}+\text{fn}} \end{aligned}$$

We define F to be 0 if there are no true positives. We compute the expectations of tp's, fp's, and fn's following [12]:

$$\widehat{\text{tp}} = \sum_{i=1}^{n} \hat{p_i} d_i, \widehat{\text{fp}} = \sum_{i=1}^{n} (1-\hat{p_i}) d_i, \widehat{\text{fn}} = \sum_{i=1}^{n} \hat{p_i}(1-d_i)$$

where n is the number of documents, $\hat{p_i}$ is the estimated probability of document i being relevant and $d_i = 1$ for $\hat{p_i} >= 0.5$ and $d_i = 0$ for $\hat{p_i} < 0.5$.

Following Lewis, we can then estimate F as follows:

$$\begin{aligned} \hat{F} &= \frac{2\sum_{i=1}^{n}\hat{p_i}d_i}{2\sum_{i=1}^{n}\hat{p_i}d_i + \sum_{i=1}^{n}(1-\hat{p_i})d_i + \sum_{i=1}^{n}\hat{p_i}(1-d_i)} \\ &= \frac{2\sum_{i=1}^{n}\hat{p_i}d_i}{\sum_{i=1}^{n}(\hat{p_i}+d_i)} \end{aligned}$$

Lewis derives error bounds for this estimator in [12] and shows that, for two hypothetical data sets, it performs well if the classifier estimates the distribution of $\hat{p_i}$ over the data correctly and if $\sum_{i=1}^{n}\hat{p_i}$ and $\sum_{i=1}^{n} d_i$ are not too small. In this paper, we test the estimator empirically. The important property of $\hat{F}$ for our purposes is that it allows us to assess the quality of a classifier without a large *labeled* training set. All we need is a large *unlabeled* random sample and unbiased probability estimates. We can compute probability estimates $\hat{p_i}$, make classification decisions d_i based on these estimates, compute an estimate of F and then make a deployment decision – all without actual labels.

However, Lewis' simulations suggest that this decision procedure for AL will only produce correct results if the probability estimates are unbiased. Biased estimates will in general lead to overly optimistic or pessimistic estimates of F because the classifier over- or underestimates its confidence in making decisions.

[1] If the pool is finite, all documents will eventually be added to the training set, but relying on the finiteness of the pool is no better than exhaustive labeling of all available data.

There can be a conflict between the two goals of classification accuracy and "unbiasedness," i.e. the goal of producing unbiased probability estimates. Consider two classifiers c_1 and c_2 for a class C with prior probability $p_C = 0.5$. The classifiers assign documents to the more probable class. c_1 estimates the probability of relevance for all documents as $0.5 + \epsilon$. Then c_1 is an unbiased estimator. Its classification accuracy is 50%. c_2 estimates a probability of relevance of 0.0 (without bias) for non-relevant and 0.9 (with negative bias) for relevant documents. c_2 is a biased estimator because it underestimates its certainty by 0.1 for relevant documents, but it has higher accuracy than c_1 (100 vs. 50). A choice between c_1 and c_2 is a choice between classification accuracy and unbiasedness. Ideally, we can find a method that is optimal on both counts; but in practice we may have to trade less bias for worse classification accuracy. This tradeoff does in fact occur in the experiments below (methods C vs. LR for estimating precision).

We do not view this tradeoff as a traditional bias-variance tradeoff [7] because the learning problem (in particular, the training set) is variable. We are comparing a learning problem with a randomly selected training set (low classification accuracy, low estimation bias) with a learning problem with a different, actively learned training set (high classification accuracy, high estimation bias). This comparison is the one of interest here since we want to hold the cost of classification, which corresponds to the number of labeling decisions, constant.

There is a large body of research about methods that produce unbiased probability estimates (e.g., logistic regression) versus those that do not (e.g., Naive Bayes, [5]). However, the problem at hand is complicated by the fact that our training set is 1) small and 2) biased.

In the following sections, we look at 5 different classification methods and their ability to provide unbiased probability estimates in the service of our decision procedure for AL. These methods are also compared with leave-one-out estimation.

4. EXPERIMENTS

The first half of the RCV1 corpus [15] (400,001 documents) was used as the experimental testbed.[2] This first half was randomly split into POOL and EVAL. Seed and query documents are drawn from POOL, EVAL is used for evaluation.

In the first set of experiments linear SVMs [29] were used because they are generally viewed as having optimal or close to optimal performance in text categorization [31].

Each document is represented as a stemmed, tf-idf-weighted word frequency vector, using the formula $(1 + \log \mathrm{tf}) \cdot \log \frac{N}{\mathrm{df}}$ for $\mathrm{tf} > 0$ where N is the number of documents. Document vectors are normalized to unit length. A stop list of common words was used. Using this representation, the document vectors have 276,727 dimensions. Note that in contrast to other AL studies, feature selection was not necessary.

We use uncertainty sampling as proposed by [14], which, in each iteration, selects the most uncertain document for labeling. The seed set for AL consisted of 5 positive and 5 negative documents that were randomly selected from POOL. 100 iterations of AL were performed consisting of computing an SVM model ($M_i, 0 \leq i \leq 99$) on the labeled set, labeling the document with score closest to 0, and adding it to the labeled set. One last model, M_{100}, was then computed on the labeled set of size 110.

The number of 100 iterations was chosen because the expert's time per category is often limited due to cost constraints. If an expert (often a highly paid and specialized employee who is needed elsewhere) can label 2 examples per minute, then labeling 110 examples for 43 categories would take roughly one week. When highly uncertain long documents (as opposed to short titles) are judged for extended periods of time, then a speed of greater than 2 per minute is hard to exceed, especially if high labeling accuracy is required (as is the case in AL).

One set of 10 categories was selected randomly from each of 4 frequency ranges of n ($[10^1, 10^2)$, $[10^2, 10^3)$, $[10^3, 10^4)$, and $[10^4, 10^5)$) where n is the number of positive documents in POOL and EVAL. All categories with $n > 10^5$ (a total of 3) were included. Categories with $n < 10^1$ were excluded. The evaluation set therefore contains 43 categories. By construction, the set contains a mix of "small" and "large" categories and is in that respect a good model of many practical text classification tasks. 5 trials of AL were run for each of the 43 categories.

To determine the optimal level of performance, one set of SVM models was trained on the entire pool (which was treated as labeled in this case). Table 1 compares the performance of this optimal model (column "optimal") with M_{100}. Performance of M_{100} is about 9% worse on average (56 vs. 65). All performance numbers in Table 1 were computed on EVAL.

To model a realistic AL scenario, we needed a mix of learnable and unlearnable categories. We confirmed post-hoc that our selection procedure achieved this. The median F on EVAL after 100 AL iterations is 65. 12 categories had an optimal F score of less than 50, which qualifies the category as not having been learned in most practical settings we are familiar with.

The experimental system was implemented in python. It uses svmlight [11] for SVM computations (with default parameters for linear SVMs) and R for (non-regularized) logistic regression.

4.1 Leave-one-out estimation

Leave-one-out (LOO) estimation is commonly used for accuracy estimation. 110 SVM models were trained on the different subsets of size 109 of the labeled set and then applied to the remaining labeled document. The 110 decisions were recorded and evaluated using F. The average estimation error of LOO was -4, the average absolute estimation error was 19 (see Table 1, column Δ LOO). For each category, the standard deviation of the (non-absolute) error over 5 trials was computed. The average of these 43 standard deviations was 6, indicating moderate variability. Sigmas for all other non-absolute means in Table 1 were similarly computed to show that variability across the 5 runs was moderate in general (with the exception of ΔS).

The magnitude of the estimation error (19) makes LOO unusable in practice. To illustrate this point, we estimated the expected error of accepting a classifier with performance $\hat{F}$ estimated by LOO and true (unknown) performance F

[2] We intend to use the second half as a temporally separated data set for research not discussed here.

		optimal	M_{100}	Δ LOO	S	Δ S	C	Δ C	SM	Δ SM	CM	Δ CM	LR	Δ LR
F	$\overline{q_i}$	65	56	−4	57	+23	28	+55	61	+22	61	+34	58	-39
	$\overline{\sigma_c}$		3	6	8	17	3	5	4	6	3	5	4	4
	$\overline{\lvert q_i \rvert}$			19		30		55		27		34		39
P	$\overline{q_i}$	82	78	−15	69	+25	80	+2	75	+16	80	+16	73	+7
	$\overline{\sigma_c}$		5	6	12	12	4	3	7	7	5	6	7	6
	$\overline{\lvert q_i \rvert}$			18		26		5		19		16		12
R	$\overline{q_i}$	57	48	−2	59	+19	19	+65	57	+21	54	+40	53	-41
	$\overline{\sigma_c}$		3	6	6	17	2	5	5	8	4	5	3	3
	$\overline{\lvert q_i \rvert}$			18		27		65		32		40		41

Table 1: Evaluation of F, Precision (P), Recall (R) and their estimation errors. Values are given as percentages. Columns refer to "Optimal" (SVM trained on the entire POOL, all other methods are trained on the actively learned training set), M_{100} (SVM), S (method Simple), C (method Committee), SM (method Simple-Mixed), CM (method Committee-Mixed), and LR (logistic regression). Δ columns give the error for the corresponding method ($\hat{F}_i - F_i$, $\hat{P}_i - P_i$, and $\hat{R}_i - R_i$). Δ LOO is the error of LOO for estimating the performance of M_{100}. Depending on the row, q_i refers to error in estimating F, P, and R (Δ columns) or to F, P, and R (remaining columns). $\overline{q_i}$ and $\overline{|q_i|}$ are averaged over 43 categories × 5 trials (except for the $\overline{q_i}$ in column "optimal", which average over 43 classes since the optimal SVM is independent of the seed set). $\overline{\sigma_c}$ is the average of 43 standard deviations (one per class), each computed on a set of 5 q_i.

using threshold θ on $\hat{F}$ by

$$\begin{aligned} E[L(\hat{F}, F|\theta)] &= P(\hat{F} < \theta | F > \theta) P(F > \theta) \\ &\quad + P(\hat{F} > \theta | F < \theta) P(F < \theta) \end{aligned}$$

where L is the (unit) loss that occurs when the decision is wrong. Assuming that the LOO estimates $\hat{F}$ are normally distributed (with parameters estimated from the 5 trials for each of the 43 categories), thresholding at 0.75 has a 35% chance of making a mistake over all the categories.

LOO is effective if the labeled set is random and large. But if the labeled set is biased and small, as in our case, LOO does not provide good estimates and therefore does not lend itself to making reliable deployment decisions.

The premise of Lewis' method is that probability estimates are unbiased. We test four different strategies for computing unbiased estimates: a hybrid model that converts scores into probabilities (method S), bagging (models C and CM), a model that is trained on unlabeled data (model SM) and multivariate logistic regression (model LR). Models S, C and LR are standard models commonly used in text classification (see references below). Method SM (and also CM) were designed to use unlabeled data for the specific purpose of producing unbiased probability estimates. Most work in machine learning that exploits unlabeled data is instead focussed on improving classification accuracy.

4.2 SVM models

The first model is a simple hybrid SVM classifier. M_{100} is applied to the final labeled set of size 110. A logistic model is then fit on the 110 scores as predictors and the known labels as responses. We call this model "simple" (S). This type of conversion of scores into probabilities is common in text classification [14, 20].

Classification and estimation results for method S are shown in columns S and Δ S of Table 1. Classification accuracy is similar to that of M_{100} (57 vs. 56). Estimation error is larger than for LOO: average absolute error is 30 (vs. 19 for LOO). The estimation bias (Δ S) is strongly positive: +23.

The reason for this positive bias is the bimodal distribution of SVM scores that the logistic model is trained on: they are either close to -1 or 1. As a result, most estimated probabilities of relevance are close to 0 or 1: 96% of probability estimates are in $[0.0, 0.1] \cup [0.9, 1.0]$. In other words, the classifier is too sure of itself due to a training set with few documents in the middle range between clearly non-relevant and clearly relevant.[3]

For the bagging [2] classifier (method C), we follow [21]. 5 SVMs were trained on subsets of 109 of the labeled set. The 5 SVMs were applied as a voting committee to EVAL and the probability of class membership was then computed as the proportion of yes votes. In initial experiments we found that uncorrelated committees perform best. To find 5 uncorrelated methods we selected from the unlabeled pool a subset C consisting of the 1000 documents whose M_{100} scores were closest to 0 (500 with positive scores, if available, and the rest with negative scores). The similarity s between two models was then defined as the correlation of their scores on C. With respect to a set of available models A and a subset of selected models U, we define the minimally similar model $D_A(U)$ to be $\mathrm{argmin}_{L_j \in A-U} \max_{L_k \in U} s(L_j, L_k)$. We start with the set $A = \{L_i | 1 \leq i \leq 110\}$ where L_i is the model trained on the set of 110 labeled documents minus document i. We then selected $\{D_A^i(\{L_{110}\}) | 0 \leq i \leq 4\}$ as our committee of 5. Unfortunately, model C's classification and estimation results are poor, in fact the average error of +55 was the worst in any of the experiments performed. Bagging does not seem to produce accurate probability estimates for small biased training sets.

The third method, SM, exploits the distribution of unlabeled data for more accurate estimates. The basic idea is to guess the labels of unlabeled documents in the pool based on their distances from the separating hyperplane.We defined the uncertainty margin as all points at a distance of at most d from the separating hyperplane computed by

[3] We ran the same experiment with disjoint training sets for SVM and logistic regression (under a cross-validation regime) on the hypothesis that overtraining might exacerbate bias in estimation. However, there was no significant increase in the accuracy of estimates of F.

M_{100}. For the experiment we chose $d = 0.25$. An unlabeled document with score s in POOL was "artificially" labeled as true for $s > d$, as false for $s < -d$ or assigned a class membership probability of $0.5-\epsilon$ for $-d \leq s \leq 0$ and $0.5+\epsilon$ for $0 < s \leq d$. We chose $\epsilon = \frac{1}{6}$ because the four probabilities $\{\frac{n}{3}, 0 \leq n \leq 3\}$ are the simplest way of distinguishing the four cases: certainly positive, uncertain tending positive, uncertain tending negative, certainly negative.

A logistic model was then trained on the SVM scores of M_{100} as predictors and the union of true labels (for the labeled set) and artificial labels as responses. We call this model "simple-mixed" (SM) since it is trained on a mix of labeled and unlabeled data. The model's performance for F is similar to M_{100}, with the average being higher (61 vs 56, column SM). The absolute error in estimating F is worse than LOO (27 vs. 19, column Δ SM).

The last SVM method, "committee mixed" (CM), combines information from the 5 models in method C and the unlabeled pool in method SM. The 5 committee models were selected to be as diverse as possible in the hope that the amount of disagreement among models can be converted to an unbiased probability estimate. As an indicator of confidence in the prediction of the models, we simply use the average of the 5 scores. The "artificial" labeling of the unlabeled pool is performed by method C so that documents are assigned a class membership probability in $\{0.2 * n | 0 \leq n \leq 5\}$. A logistic model is trained on the averages of the 5 SVM scores as predictors and the union of true labels (for the labeled set) and method C predictions (for the unlabeled pool) as responses. In classification, the 5 SVM scores are averaged and the logistic model is then applied to this average. Classification accuracy of method CM is better than M_{100} (61 vs. 56, column CM), but estimation of F is worse than for LOO (average absolute error of 34 vs. 19, column Δ CM).

4.3 Regularized logistic regression

SVMs are optimized with respect to optimal discrimination without regard to producing accurate probability estimates. The multi-stage procedures that convert scores into probability estimates are not guaranteed to be optimal. We therefore also tested regularized logistic regression (LR) as a method that outputs unbiased probability estimates if certain assumptions hold. We used the BBR package [8] to train a logistic classifier on the final training set of 110 labeled documents.

The average F of LR was 58 (column LR). The bias of the estimation error for precision (+7, column Δ LR) is smaller than for any of the other reasonably performing methods (method C is an exception due to its poor F). However, the magnitude of the absolute error of F (39) is larger than for the SVM-based methods (again, except for method C).

4.4 Support vector machine AL

All AL methods are by construction biased: Training set examples are selected non-randomly in order to create a maximally informative training set for a given number of labeling decisions. It is however possible that some AL methods construct training sets with a higher bias than others. To investigate a possible dependency of bias on AL method, we ran a final set of experiments with support vector machine active learning (SVMAL [27]; here, LIBSVM [3] was used). SVMAL trains two SVM models for each member of the pool in each iteration, thus making a repetition of our large experiment with SVMAL infeasible. We instead ran a smaller experiment with 5000 documents in the pool and 20 classes. The average absolute errors were 36 and 38 for the MaxMin Margin and MaxMin Ratio approximations of SVMAL, respectively, vs. 37 for uncertainty sampling. This indicates that SVMAL-based accuracy estimates of F are too error-prone to be usable in reliable deployment decisions of classifiers.

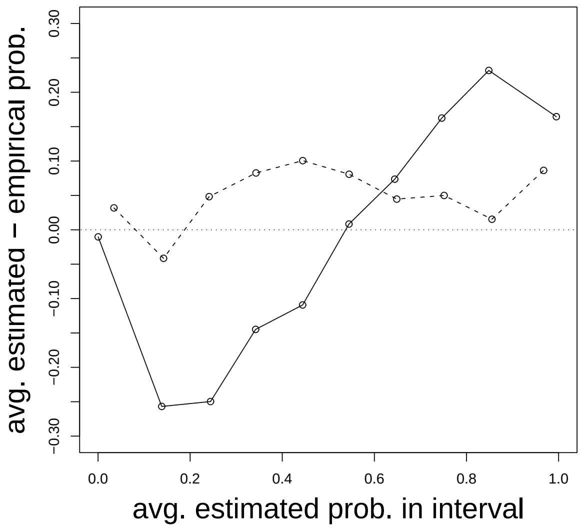

Figure 1: Estimation error of methods CM (solid) and LR (dashed). Example: the true probability of relevance for documents with LR estimates in the interval [.9,1] was .881, the average estimate was .967, so the estimation error was .967-.881=.086. This is represented as the last data point on the dashed line: (.967,.086).

5. ANALYSIS OF EXPERIMENTAL RESULTS

We use the sign test [26] for significance testing. It is a weak test, but it turned out to be sufficiently sensitive for our purposes. The number of paired samples is 215 (43 categories × 5 trials). We report the two-tailed significance level p. The main result of the experiments is that Lewis' estimator does not estimate F better than LOO. The other 5 methods perform worse than LOO, and all but method SM do so significantly ($p < 10^{-5}$). We performed a detailed analysis for methods CM and LR. Figure 1 shows that CM fails because it overestimates for positive decisions ($\hat{p} > 0.5$), thus increasing the precision estimate, and underestimates for negative decisions ($\hat{p} < 0.5$), thus increasing the recall estimate. Both overestimating for $\hat{p} > 0.5$ and underestimating for $\hat{p} < 0.5$ amount to an overestimation of the classifier's certainty that its decision is correct and therefore lead to large positive errors. Logistic regression also has a positive bias for positive decisions (dashed line), albeit a smaller one; but it overestimates probabilities for negative decisions, thus underestimating recall. (The "neg-

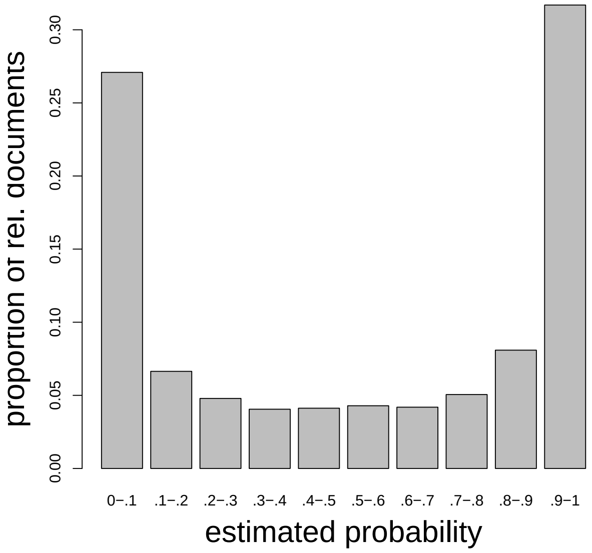

Figure 2: Distribution of relevant documents with respect to estimated probability. Each bar corresponds to the proportion of relevant documents that receive probability estimates in the corresponding range on the x-axis. 27% of relevant documents receive LR probability estimates of less than 0.1, indicating the presence of missed clusters.

ative" interval [0.1,0.2] in Figure 1 only contains 17% of all documents, so that the positive bias for the other intervals dominates.) This explains why logistic regression has a large underestimation error (-41) for recall.

For methods S, CM and SM, final probabilities are produced by a univariate logistic regression on a distribution of scores with most values close to -1 or 1 (since the small actively learned training sets can easily be linearly separated in high-dimensional space). As a result, probabilities are close to 0 or 1 (i.e., in $[0.0, 0.1] \cup [0.9, 1.0]$). This is true for more than 99% of probability estimates for method CM. In contrast, the multivariate logistic regression employed by method LR produces many "uncertain" estimates: only 71% of estimates are close to 0 or 1. This uncertainty results in less optimistic precision estimates and pessimistic recall estimates.

The experiments suggest that it is hard, if not impossible, to estimate recall correctly based on ensemble methods (C, CM) or methods exploiting the distribution of unlabeled data (SM, CM). LOO performs best with an absolute error of 18. The other 5 methods perform significantly worse ($p < 10^{-5}$). An error of 18 or worse will make the decision of whether a classifier can be deployed or not subject to high error when relying on LOO estimates. Recall estimation is hard because of the missed cluster effect discussed earlier. In the absence of information about whether there are unexplored parts of the space with relevant documents, an accurate assessment of recall is impossible.

In addition to missed clusters there are however other conceivable causes for incorrect probability estimates for true and false negatives. In particular, a difficult decision boundary could also give rise to poor recall estimation. In that case, true and false negatives with incorrect estimates would be located close to the decision boundary. Figure 2 shows that instead, a large number of documents are located *far* from the decision boundary (i.e., they receive LR estimates close to 0). This indicates that the main culprit for poor recall estimation is indeed the missed cluster effect.[4]

	S	SM	C	CM	LR
LOO	$\ll$	$\approx$	$\gg$	$\gg$	$\gg$
S		$\gg$	$\gg$	$\gg$	$\gg$
SM			$\gg$	$\gg$	$\gg$
C				$\ll$	$\ll$
CM					$\gg$

Table 2: Sign tests comparing error of precision estimates for 6 methods. $\ll$/$\gg$ indicate a level of significance of $p < .01$. $\approx$ indicates lack of significance.

Despite the difficulty of recall estimation, methods based on Lewis' estimator should in principle be able to compute unbiased estimates for the densely sampled space around the decision boundaries and hence be able to estimate precision accurately. Indeed, three of the methods (C, CM, and LR) are more accurate than LOO for estimating precision (see Table 2). LR beats all methods but C. However, C is the least accurate classifier: its F at 28 is less than half of optimal. This is an example of the estimation accuracy vs. unbiasedness tradeoff discussed earlier. C's precision estimates are unbiased, but it achieves this by only assigning documents to the category that it is absolutely certain about. As a result, it can estimate the precision it achieves better than the other methods.

We conclude from these results that recall cannot be estimated accurately, but that probabilistic methods like multivariate logistic regression can, in principle, estimate precision correctly. While 12 is still a large absolute error, we have used the BBR system out-of-the-box without any tuning. Additional optimization for accurate estimation of precision should improve this result.

6. RELATED WORK

[22] present a stopping criterion for AL, but they do not address the issue of accuracy estimation. Having achieved optimal performance is not a satisfactory deployment criterion if that optimal performance is below minimum acceptability. Also, the arguments with respect to recall apply to their proposed stopping criterion as well. No stopping criterion can tell us for sure that recall has reached an optimal level. Decidability is therefore not addressed.

There are other algorithms besides SVMAL that perform

[4] For LOO, over- and underestimation of recall almost balance out across classes (bias is -2). The missed cluster effect explains cases of positive bias. Negative bias can occur for highly non-redundant training sets. A document that has no close neighbors is likely to be misclassified in LOO. Rank correlation of [bias] and [number of documents in the training set whose closest neighbor had a cosine similarity of more that 0.15] (a measure of redundancy) was 0.54 ($p < 0.001$). This correlation supports the hypothesis that non-redundancy and missed clusters are counteracting effects in LOO, but further research on this issue is necessary.

somewhat better than uncertainty sampling, but are also computationally more expensive (e.g., [24, 6, 10]). We chose the computationally most efficient AL method, uncertainty sampling, because a fast querying method is important, so that experts can be provided with the next document within a few seconds of their last judgment, even if the pool is large. Longer response times prevent concentration on the task at hand [25]. An efficient implementation of uncertainty sampling on a fast multi-processor machine meets this criterion, even for a very large pool as it was used in this paper. The pool must be large for learning small categories.

As in AL, learnability is also a problem for a combination of supervised and unsupervised learning [1, 17, 19]. We would like to extend our results to combination learning strategies.

Bagging performed reasonably well for accuracy estimation in [21]. However, the newsgroup categories used in this study are taken from non-overlapping distinct corpora, e.g., politics vs. electronics newsgroups. Reuters categorization tasks are more difficult since categories are overlapping and correspond to more subtle human categorization decisions. The difference between Naive Bayes and SVM may also play a role. The accuracy/unbiasedness tradeoff implies that it is harder to get accurate probability estimates for a classifier with optimal (or close to optimal) accuracy.

[16] find that a disagreement-based accuracy estimator performs well compared to LOO in a transductive learning setting for balanced categories. This result is not directly applicable to experiments where training and test set are drawn from the same distribution and categories are "small" (or unbalanced) as in this paper.

In this paper, we have investigated three performance measures: F, P, and R. Another important measure is utility (e.g., [32], where, as in our case, the complicating circumstance of a biased training set is investigated). We don't see any difference between accuracy estimation for F, P, and R vs. utility in principle: It seems hard to conceive of an unbiased estimator of utility that would not in turn rely on unbiased estimates of relevance. The literature on filtering has focussed on the decision $p > \theta$ where p is the probability of relevance and θ is the filtering threshold. However, estimates of relevance can be strongly biased (and therefore unsuitable for accuracy estimation) even if they "answer" the question "$p > \theta$?" correctly. This question has not been investigated in filtering as far as we know.

7. DISCUSSION

7.1 Unbiased estimation

We have argued that learnability and decidability are critical issues in practical text classification and investigated unbiased estimation as a possible solution. The experimental results suggest unbiased estimation is hard. In fact, we believe that there is no general solution to unbiased estimation for small and biased sets as they are produced in AL.

This is partly due to the missed cluster problem. If there are important parts of the representational space that have no representation in the small and biased AL sample, then available information is simply not sufficient for estimating a model that would be consistent with the true distribution of relevant documents.

Another possible solution to accuracy estimation would be to correct the bias. [13] quotes Catlett as suggesting stratification as a bias correction strategy. Stratification is effective if we have an independent characterization of the strata, for example their true relative frequencies. However, it is not clear with respect to what we can stratify in the estimation problem at hand. Stratification with respect to the document space is out of the question because of its high dimensionality. An alternative that we have tested experimentally is to stratify with respect to the SVM scores or probability estimates that are output by the classifiers for the AL set. However, these attempts did not produce better estimates of true performance than the Lewis' estimator used here. The problem is again the small size of the sample which does not contain enough information to reliably estimate the density of relevant documents, especially in the "low probability" part of the space (low probability according to the actively learned classifier).

If learnability and decidability are serious problems, why have they not been addressed in previous work on AL? One reason is that most experimental work has been done on a few text collections with categories that have a straightforward correspondence between word distribution and category membership. For example, most of the categories in the 20 newsgroups corpus are defined by a set of words that occur together, words such as "graphics", "2D", "3D", "circle", "bezier" for comp.graphics. If significant words cooccur with each other enough, then AL can discover them one after the other. Cases like the commodities cluster for Australia will not occur. Recall is likely to be high after a small number of iterations and those parts of the space that contain relevant documents will be sufficiently densely sampled to compute usable estimates.

The importance of decidability has not been realized because in most AL experiments the optimal performance for a classifier is high and classes are learnable. In this type of scenario, the typical number of iterations used in the literature (50–100) is sufficient to learn the category with acceptable performance. But this is an unrealistic assumption for practical text classification. In most cases, we don't know what the best possible level of performance is and many practical applications have a mix of learnable and unlearnable categories.

We have shown that none of the methods tested here are estimators that would produce a good estimate of R, and hence F. The reason is that their probability estimates are highly biased. We are not aware of other methods that produce unbiased estimates for small and biased training sets. Our conclusion is that, except for those cases where a large random training set is available in practical text classification, Lewis' estimator cannot be used to address the problems of learnability and decidability in AL.

7.2 Practical text classification

The alternative to Lewis' estimator is evaluation on a random sample. Indeed, most publications on text classification evaluate experiments with respect to a held-out random sample. They do not address the problem that such a held-out random sample is usually not available in practical text classification. In particular, it is not available in AL. AL has therefore no built-in method for evaluating the success of learning.

This means that neither of the two avenues for determining whether a category has been learned is available. Neither random sampling nor unbiased estimation (and then using Lewis' estimator) is an option that is available as a generic method in practical text classification.

However, if we are willing to make additional assumptions or expend significant additional resources, then the dilemma posed here can be overcome in certain situations. We conclude by characterizing some of the scenarios in which practical text classification problems can be solved even though there is no large randomly sampled training set available.

- There is some kind of extrinsic validation of the classifier. For example, the cost of a non-performing classifier for the business may be low. One can therefore deploy actively learned classifiers. Over time, a subset will be identified as underperforming by means of the extrinsic validation. These classifiers can then be retired.

- For categories of intermediate size a strategy of mixed random sampling and AL may be successful. AL makes sure that the decision boundaries are learned well. Random sampling discovers missed clusters. We can view this approach as an exploration/exploitation tradeoff where random sampling serves the purpose of exploration and AL the purpose of exploiting each labeling decision for maximal information about the decision boundaries.

- If the category is large enough one may be able to completely dispense with AL. The arguments in this paper mainly apply to categories with small population rates. Random sampling is a good alternative to AL for categories with large population rates.

- Some tasks only require good precision. High recall is less important. Web search is the classical example. It is often evaluated by precision at a certain cutoff – a measure that requires a minimum level of recall, but is otherwise dominated by precision. As we argued above, there is hope that methods for estimating precision with reasonable accuracy can be developed. AL would then be a practical strategy for "precision-only" tasks.

- The most promising avenue for addressing the dilemma we face here is to avail ourselves of more domain knowledge. In many cases, the classification problem is of a much lower dimensionality than the high-dimensional term space typically used for representation. If domain knowledge can guide us towards finding this lower-dimensional space, then different solutions become available. For example, stratification may become feasible if dimensionality can be reduced sufficiently. Many text categories are centered around a concept that is consistently expressed with a small set of terms such as the terms "Australia" and "Australian" in the case of the category "Australia". If we know that there is such a small set of terms and if we can identify it, then the dimensionality of the space that needs to be sampled to assemble a representative set of strata can be reduced. As a result, reliable estimates of performance can be computed by way of stratification.

We conclude from our experiments that AL is not a general solution to the problem of practical text classification. The key to practical text classification is to bound the area of applicability of AL, the method of choice for creating the training set in supervised text classification. It is therefore important to characterize scenarios where AL is applicable and the problems of learnability and decidability can be addressed.

Acknowledgments. We are grateful to Reuters for RCV1 and to Byron Dom, Omid Madani, Amir Najmi, Helmut Schmid, Jason Utt, Tong Zhang and the anonymous reviewers for comments.

8. REFERENCES

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[2] L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[4] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Mach. Learn.*, 15(2):201–221, 1994.

[5] P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3):103–130, 1997.

[6] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Mach. Learn.*, 28(2-3):133–168, 1997.

[7] J. H. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

[8] A. Genkin, D. D. Lewis, and D. Madigan. Large-scale bayesian logistic regression for text categorization. In preparation, 2005.

[9] D. Haussler. Probably approximately correct learning. In *AAAI*, pages 1101–1108, 1990.

[10] V. S. Iyengar, C. Apte, and T. Zhang. Active learning using adaptive resampling. In *SIGKDD*, pages 91–98. ACM Press, 2000.

[11] T. Joachims. *Learning to Classify Text using Support Vector Machines.* Kluwer, 2002.

[12] D. D. Lewis. Evaluating and optimizing autonomous text classification systems. In *SIGIR*, pages 246–254, 1995.

[13] D. D. Lewis. Training text classifiers by uncertainty sampling. Manuscript, AT&T Labs, 2001.

[14] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *SIGIR*, pages 3–12, 1994.

[15] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, 2004.

[16] O. Madani, D. M. Pennock, and G. W. Flake. Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. In *NIPS*, pages 873–880, 2004.

[17] A. K. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In

ICML, pages 350–358, 1998.

[18] T. M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[19] I. Muslea, S. Minton, and C. A. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *ICML*, pages 435–442, 2002.

[20] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in large margin classifiers*, pages 61–74, 2000.

[21] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448, 2001.

[22] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *ICML*, pages 839–846, 2000.

[23] F. Sebastiani. Machine learning in automated text categorization. *ACM Comp. Surveys*, 34(1):1–47, 2002.

[24] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *COLT*, pages 287–294, 1992.

[25] S. L. Smith and J. N. Mosier. Guidelines for designing user interface software. Technical Report ESD-TR-86-278, MITRE, 1986.

[26] G. W. Snedecor and W. G. Cochran. *Statistical methods.* Iowa State University Press, 1989.

[27] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2001.

[28] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, London, 1979. Second Edition.

[29] V. N. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer, Berlin, 1982.

[30] Y. Yang. Sampling strategies and learning efficiency in text categorization. In *AAAI Spring Symposium on Machine Learning in Information Access*, pages 88–95, 1996.

[31] Y. Yang and X. Liu. A reexamination of text categorization methods. In *SIGIR*, pages 42–49, 1999.

[32] Y. Zhang and J. P. Callan. Maximum likelihood estimation for filtering thresholds. In *SIGIR*, pages 294–302, 2001.