Inductive Transfer for Text Classification using Generalized Reliability Indicators

Paul N. Bennett PBENNETT@CS.CMU.EDU
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Susan T. Dumais SDUMAIS@MICROSOFT.COM
Eric Horvitz HORVITZ@MICROSOFT.COM
Microsoft Research, One Microsoft Way, Redmond, WA 98052 USA

Abstract

Machine-learning researchers face the omnipresent challenge of developing predictive models that converge rapidly in accuracy with increases in the quantity of scarce labeled training data. We introduce Layered Abstraction-Based Ensemble Learning (*LABEL*), a method that shows promise in improving generalization performance by exploiting additional labeled data drawn from related discrimination tasks within a corpus and from *other* corpora. *LABEL* first maps the original feature space, targeted at predicting membership in a specific topic, to a new feature space aimed at modeling the reliability of an ensemble of text classifiers. The resulting abstracted representation is invariant across each of the binary discrimination tasks, allowing the data to be pooled. We then construct a context-sensitive combination rule for each task using the pooled data. Thus, we are able to more accurately model domain structure which would not have been possible using only the limited labeled data from each task separately. Using several corpora for an empirical evaluation of topic classification accuracy of text documents, we demonstrate that *LABEL* can increase the generalization performance across a set of related tasks.

Keywords

Inductive transfer, ensemble methods, classifier re-use, reliability models

1. Introduction

Given the typical scarcity of labeled data for building predictive models, the Machine Learning community has pursued methods which make use of information sources beyond the labeled data associated with a pure supervised-learning framework. An example of research in this arena is multitask learning (Caruana, 1997). In multitask learning, additional information for building models comes in the form of labels for related functions which can be learned over the same input. Although such additional labels are typically unavailable at prediction time, results have demonstrated that generalization performance can be improved on the primary task by learning to predict the new variables in addition to the output variable of interest.

We are interested in improving the performance of predictive models for cases where we have inadequate amounts of labeled training data. In contrast to multitask learning, we seek to leverage labeled data from related problems over different examples to enhance the final model used in prediction. Problems related to this challenge have been termed *classifier re-use* (Bollacker & Ghosh, 1998) or *knowledge transfer* (Cohen & Kudenko, 1997). We introduce a new approach to the challenge that hinges on mapping the original feature space, targeted at predicting membership in a specific topic, to a new feature space aimed at modeling the reliability of an ensemble of text classifiers.

The approach, which we call **L**ayered **A**bstraction-**B**ased **E**nsemble **L**earning (*LABEL*), has two subcomponents. First, a set of classifiers are trained on each task according to the standard supervised learning framework; a problem or task consists of determining binary membership in a specific topic. Then, we build a context-sensitive ensemble model using these classifier outputs and a set of *reliability-indicators*—a set of features that provide an abstraction of discriminatory context appropriate for modeling classifier

Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data, Washington DC, 2003.

reliability. We thus abstract away the problem of predicting specific class membership to that of predicting the reliability of a set of classifiers for a given class. As a result, both the input features and their relationship to the class variable are the same at the metalevel; this enables the simultaneous use of all the data as a model bias across the entire set of tasks.

We first review related work and our previous work that demonstrated robust gains on a task-by-task basis across a variety of topic classification problems using reliability indicators. Then we describe in detail how the *LABEL* methodology generalizes the earlier work by providing a means for using data across tasks. Finally, we present an empirical analysis of this methodology applied to text classification and summarize the strengths and weaknesses of the approach.

2. Related Work

Before moving on, we shall highlight a few of the research veins that have tackled issues related to knowledge transfer. Caruana (1997) presents an approach to and analysis of multitask learning when the n function-approximation tasks are over the same input (*i.e.*, a labeled example consists of $x_1, \ldots, x_m$ data attributes and the values for this example of the n functions to be learned $f_1(\vec{x}), \ldots, f_n(\vec{x})$). In this analysis, the main concern is generalization performance for one particular f_i, the primary problem. Likewise, The Curds & Whey approach proposed by Breiman and Friedman (1995) solves a similarly formulated problem but attempts to minimize the squared error across all of the n functions instead of placing emphasis on one task.

Thrun and O'Sullivan (1996) present methods for identifying related tasks and sequentially transferring knowledge when using a nearest-neighbor classifier. These methods are applicable when the input has the same representation across tasks. Both Thrun & O'Sullivan's and Breiman & Friedman's work could be applied to the problem here after transforming the data to the *LABEL* representation that generalizes across tasks.

Cohen and Kudenko (1997) perform an analysis of classifier re-use and sequential knowledge transfer in information filters for text documents. This work showed that significant improvements could be introduced when the classifiers were constructed to primarily model features positively correlated with the topic (*i.e.*, word *presence* that is positively correlated with being *In-Topic*). However, the method also relies on the new task and the old task sharing significant overlap in the underlying concept to be learned.

Bollacker and Ghosh (1998) present a novel mechanism for classifier re-use where a classifier is constructed for each of a set of support tasks that are later used in predictions for a primary task. The final classification is selected by predicting the same class as the training data item (from the primary task data) that has the most similar prediction pattern using the support classifiers. Since each support classifier is applied to examples from every task, the input representation for each of the related tasks must be the same. Additionally, the scheme, like error-correcting output coding (Dietterich, 2000), relies more on an assumption that the extra-task labels will serve as a natural encoding for the data rather than other re-use mechanisms that specifically bias models or build representations of domain knowledge.

3. Problem Approach

In distinction to prior efforts, we introduce a representation that is semantically coherent across tasks. Such semantic coherence facilitates the use of standard methods of inductive transfer.

In earlier work, we developed a classifier combination procedure that hinged on learning and harnessing the *context-sensitive* reliabilities of different classifiers (Bennett, Dumais, and Horvitz, 2002; 2003). We found that the *reliability-indicator* methodology is useful in the arena of text classification for providing context-sensitive signals about accuracy that can be used to weave together multiple classifiers in a coherent probabilistic manner to boost overall accuracy. [1]

We first discuss our formulation of reliability-indicators that lays the groundwork for both a standard task-specific metaclassifier approach and its extension to multi-task, multi-dataset learning.

3.1. Reliability Indicator Variables

Previous approaches to classifier combination have typically limited the input at the metalevel to the output of the classifiers (Ting & Witten, 1999) and/or the original feature space (Gama, 1998). Since a classifier rarely is the best choice across a whole domain, an intuitive alternative is to identify the document-specific context that differentiates between regions where a base classifier has higher or lower reliability.

We address the challenge of learning about the reliability of different classifiers in different neighborhoods of the classification domain at hand by introducing variables referred to as *reliability indicators* which represent the "discriminatory context" of a specific document. A reliability indicator is an evidential distinction with states that are linked prob-

[1] Throughout the remainder of this paper, we will give concrete examples in terms of text classification, however, the general approach promises to be applicable for predictive modeling outside of text classification.

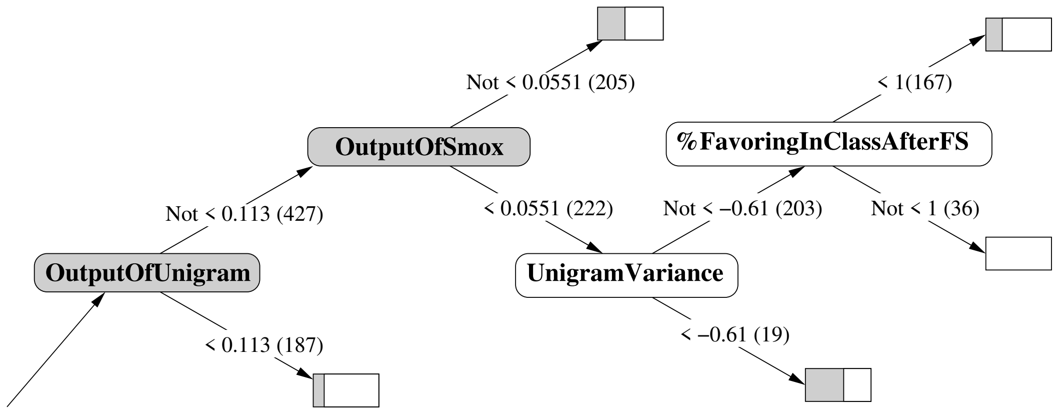

Figure 1. Portion of decision tree, learned by *STRIVE-D* for the *Business & Finance* class in the MSN Web Directory corpus, representing a combination policy at the metalevel that considers scores output by classifiers (dark nodes) and values of indicator variables (lighter nodes).

abilistically to regions of a classification problem where a classifier performs relatively strongly or poorly.

The reliability-indicator methodology was introduced by Toyama and Horvitz (2000) and applied initially to the task of combining multiple machine-vision analyses in a system for tracking the head and pose of computer users. The value of the methodology in vision motivated us to explore the analogous application of the approach for representing and learning about the reliability of text classifiers. For the task of combining classifiers, we formulate and include sets of variables that hold promise as being related to the performance of the underlying classifiers. We consider the states of reliability indicators and the scores of classifiers directly, and, thus, bypass the need to make ad hoc modifications to the base classifiers. This allows the metaclassifier to harness the reliability variables if they contain useful discriminatory information, and, if they do not, to fall back in a graceful manner on outputs of the base classifiers.

As an example, let us consider three types of documents where: (1) the words in the document are either uninformative or strongly associated with one class; (2) the words in the document are weakly associated with several disjoint classes; or (3) the words in the document are strongly associated with several disjoint classes. Classifiers (*e.g.*, a unigram model) will sometimes demonstrate different patterns of error on these different document types. If we can characterize a document as belonging to one of these model-specific failure types, then we can assign the appropriate weight to the model's output for this kind of document. We pursued the formulation of reliability indicators that capture different association patterns among words in documents and the structure of classes under consideration. We seek indicator variables that allow us to learn context-sensitive reliabilities of classifiers, conditioned on the observed states of the variable in different settings.

To highlight the approach with a concrete example, Figure 1 shows a portion of the type of combination function we can capture with the reliability-indicator methodology. The nodes on different branches of a decision tree include the values output by base classifiers, as well as the values of reliability indicators for the document being classified. The decision tree provides a probabilistic, context-sensitive combination rule indicated by the particular relevant branching of values of classifier scores and indicator variables. In this case, the portion of the tree displayed shows a classifier-combination function that considers thresholds on scores provided by a base-level linear SVM (*OutputOfSmox*) classifier and a base-level unigram classifier (*OutputOfUnigram*), and then uses the context established by reliability-indicator variables (*UnigramVariance* and *%FavoringInClassAfterFS*) to make a final decision about a classification. The annotations in the figure show the threshold tests that are being performed, the number of examples in the training set that satisfied the test, and a graphical representation of the probability distribution at the leaves. The likelihood of class membership is indicated by the length of the bars at the leaves of the tree.[2]

The variable *UnigramVariance* represents the variance of unigram weights for words present in the current document. The intuition behind the formulation of this reliability-indicator variable is that the unigram classifier would be accurate when there is low variance in weights. The variable *%FavoringInClassAfterFS* is the percentage of words (after feature selection) that occur more often in documents within a target class than in other classes. Classifiers that weight positive and negative evidence differently should be distinguished by this variable. Bennett et al. (2002; 2003)

[2]For the *STRIVE-D* excerpt shown in Figure 1 we have further normalized the metafeatures to have zero mean and unit standard deviation so most values fall between -1 and 1 as a result.

give further details about the reliability indicators used in these experiments.

The reliability-indicator variables used in our studies are an intuitive attempt at formulating states that represent influential contexts. We defined variables to represent a variety of contexts that showed promise as being predictive of accuracy—*e.g.*, the number of features present in a document before and after feature selection, the distribution of features across the positive vs. negative classes, and the mean and variance of classifier-specific weights. The experiments reported here use a total of 49 reliability indicators which were formulated by hand as an initial pass at representing potentially valuable contexts. See Bennett et al. (2002; 2003) for additional discussion of these reliability-indicators.

3.2. STRIVE

We refer to the task-specific classifier combination framework as *STRIVE* for **St**acked **R**eliability **I**ndicator **V**ariable **E**nsemble. We select this name because the approach can be viewed as extending the stacking framework by introducing reliability indicators at the metalevel. The *STRIVE* architecture is depicted graphically in Figure 2.

The *STRIVE* methodology transforms the original learning problem into a new learning problem. In the initial problem, the base classifiers simply predict the class from a word-based representation of the document. More generally, in the original problem, each base classifier outputs a distribution (possibly unnormalized) over class labels. *STRIVE* adds another layer of learning to the base problem. Reliability-indicator functions consider the words in the document and the classifier outputs to generate the reliability indicator values, r_i, for a particular document. This approach yields a new representation of the document that consists of the values of the reliability indicators, as well as the outputs of the base classifiers. The metaclassifier exploits this new representation for learning and classification. This enables the metaclassifier to employ a model that uses the output of the base classifiers as well as the context established by the reliability indicators to make a final classification.

We require the outputs of the classifiers to train the metaclassifier. Thus, we perform cross-validation over the training data, and use the values obtained while an example serves as a validation item as the input to the metaclassifier. We note that, in the case where the set of reliability indicators are restricted to be the identity function over the original data, the resulting scheme can be viewed as a variant of cascade generalization (Gama, 1998).

3.3. LABEL: Layered Abstraction-Based Ensemble Learning

Intuitively, regardless of the particular topic or source (*e.g.,* news feed, web page, etc.), topic discrimination tasks share some common structure. For example, longer documents tend to provide more information for identifying topics. Furthermore, documents containing words strongly correlated with a single topic more likely belong to that topic than documents containing words strongly correlated with several topics. Additionally, these conditions may interact with each other based on their particular values. Researchers in the field may often make similar observations after studying multiple classification problems. We seek to design a system capable of both inducing such generalizations automatically and applying them to improve the predictive performance of models.

A training corpus in text classification consists of a set of example documents labeled with each of their proper topics from a prespecified corpus-specific topic list (a document may have more than one topic). When the same representation is used for each of the binary discrimination tasks in a corpus, standard multitask learning can be used to perform classification for all of the topics in the corpus' topic list. However, standard multitask learning cannot leverage information across corpora since it would typically require knowing whether a document belongs to each of the topics from all of the corpora (where we only have in-corpus information). Additionally, the basic feature space is quite different in documents from different corpora as particular language usage varies widely. Therefore, we desire a standard representation that has the same semantics across separate tasks from both the same and different corpora.

Although *STRIVE* uses data from each task separately to build a metaclassifier for that specific task, it is straightforward to extend it to make use of labeled data across tasks. The key point is that the reliability-indicators we chose carefully abstracted away from a document's task-specific statistical regularities of word usage while maintaining the discriminatory relationship of the document's context to the task. For example, documents that come from a general topic corpus where we are trying to distinguish *Health & Fitness* from *not Health & Fitness* tend to behave very differently at the word usage distribution level than documents from a narrow financial corpus where we are trying to distinguish *Corporate Acquisitions* from *not Corporate Acquisitions*. However, in terms of the abstraction that the reliability-indicator *UnigramVariance* provides, we expect a unigram classifier to show poor reliability for a particular document from either task when *UnigramVariance* is high.

With this approach, we treat the metaclassifier as an abstraction moving the focus of the analysis from discrimi-

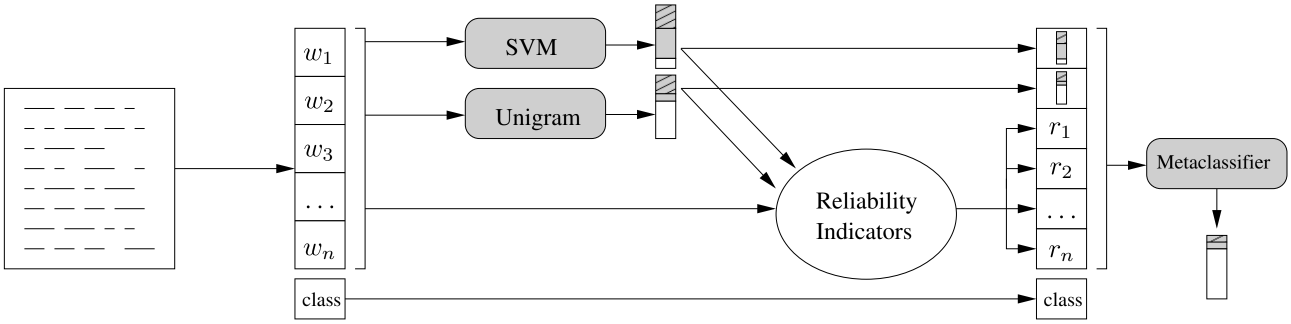

Figure 2. Architecture of *STRIVE*. In *STRIVE*, an additional layer of learning is added where the metaclassifier can use the context established by the reliability indicators and the output of the base classifiers to make an improved decision. The reliability indicators are functions of the document and the output of the base classifiers.

nating a specific topic (*e.g.*, *Corporate Acquisitions* vs. *not Corporate Acquisitions*) to the problem of discriminating topic membership (*i.e.*, *In-Topic* vs. *Out-of-Topic*). The base-level classifiers trained on a particular topic are used as the representation of topic-specific knowledge, while the metaclassifier provides information about how to leverage context across topic-classification in general.

Therefore, *LABEL*, like *STRIVE*, constructs models with the same type of combination rules as that shown in Figure 1. The differences from *STRIVE* are in the model construction procedure. After generating the metalevel data, the metafeatures are normalized to have zero mean and unit standard deviation within their particular task.[3] At this point *STRIVE* would use data from each task to separately build a metaclassifier for each task, *LABEL* departs from this by pooling all of the data together and building a single metaclassifier (with the class variable taking the value 1 if the document is *In-Topic* for the particular task and -1 otherwise).

We now give a more formal definition of the problem. For our purposes, a task is the approximation of a single binary function, $f_i(X_i) \in \{-1, 1\}$. The input domain of each of these tasks may differ; thus, X_i denotes an input example from the ith task's domain. The labeled data for each task, L_i, consists of a set of tuples $\langle \vec{x}_{i,j}, f_i(\vec{x}_{i,j}) \rangle$ (where $j = 1, \ldots, |L_i|$). Given N tasks and a performance measure *perf*, we would like an inductive learning procedure, $\mathit{Train}(i, L_1, \ldots, L_N)$, that produces a model to generate predictions for the ith task. Furthermore, we desire that our performance using all the data exceeds the performance using the data for each task separately: $\sum_{i=1}^{N} E_{P_i}[\mathit{perf}(\mathit{Train}(i, L_1, \ldots, L_N))] > \sum_{i=1}^{N} E_{P_i}[\mathit{perf}(\mathit{Train}(i, L_i))]$ where P_i is the probability distribution on the ith task. To be of practical use, the performance achieved using only labeled data from the task $E_{P_i}[\mathit{perf}(\mathit{Train}(i, L_i))]$ should be competitive with the best methods on this task, otherwise the solution is trivial (simply ensure the models using only labeled data from the task perform as poorly as possible).

[3]This is not necessary for *STRIVE*, but for *LABEL* this helps to deal with spurious statistical variance that arises from the tasks having different numbers of training examples.

Before applying the resulting model for prediction, it is desirable to specialize this single metaclassifier for each task in two ways. First, each task may have different priors, therefore these priors should be taken into account at prediction time. This can be directly accomplished by obtaining probability predictions from the metaclassifier or simply setting a different threshold for each classification task. Secondly, tasks may diverge from the average case in different ways. Thus, we may want to only retain part of the general model. The best way to address this question depends, in part, upon the choice of classification algorithm for the metaclassifier. We discuss our particular choices in Section 4.2 below.

4. Experimental Analysis

We performed an empirical analysis over standard text classification corpora to explore the effectiveness of *LABEL*. We also performed ablation experiments to elucidate how *LABEL* achieves an improvement in generalization performance. Each of the classification models use a decision threshold specific to each task. The threshold for each model and task was empirically determined over the training data.

4.1. Base Classifiers

We selected for our experiments four classifiers that have been used traditionally for text classification: decision trees, linear SVMs, naïve Bayes, and a unigram classifier.

For the decision-tree implementation, we employed the WinMine decision networks toolkit and refer to this as *Dnet* below (Microsoft Corporation, 2001). *Dnet* builds decision trees using a Bayesian machine learning algorithm (Chickering, Heckerman, and Meek, 1997; Heckerman et al., 2000). Although this toolkit is targeted primarily at building models that provide probability estimates, we found that *Dnet* models usually perform acceptably for the goal of minimizing error rate. However, we found that the performance of *Dnet* with regard to other measures is sometimes poor.

For linear SVMs, we use the *Smox* toolkit which is based on Platt's **S**equential **M**inimal **O**ptimization algorithm (Platt, 1998). A continuous model of the feature space is used.

The *naïve Bayes* classifier has also been referred to as a multivariate Bernoulli model. In using this classifier, we smoothed word probabilities using a Bayesian estimate with the word prior and smoothed class probabilities using a Laplace m-estimate.

The *unigram* classifier uses probability estimates from a unigram language model. This classifier has also been referred to as a multinomial naïve Bayes classifier. Probability estimates are smoothed in a similar fashion to smoothing in the *naïve Bayes* classifier.

Since *Smox* is the best base classifier in the experiments below, it is the only base classifier we report in summarizing our experimental results.

4.2. Metaclassifiers

As mentioned above, the inputs to the metaclassifiers are normalized to zero mean and unit standard deviation (as estimated during the training phase). The experiments reported here use a total of 49 reliability indicators which were formulated by hand as an initial pass at representing potentially valuable contexts (additional detail can be found in Bennett et al. 2002; 2003).

For these experiments, we used only a decision-tree algorithm (using Dnet) as a metaclassifier. For this reason, we refer to the primary metaclassifier implementations below as *STRIVE-D* and *LABEL-D*. We note that by comparing these two systems directly, we see the effects of *separately* building a metaclassifier per task versus building them in conjunction.

Here, we introduce one way to specialize the single metaclassifier model learned by *LABEL-D* for decision trees. Given a single metaclassifier decision tree model, instead of using the prediction at each leaf node as the aggregate distribution across tasks of *In-Topic* vs. *Out-of-Topic*, when predicting for task T, we use the estimate:

$$P(\textit{In-Topic}_i \mid \text{leaf} = l) = \frac{\textit{In-Topic}_{i,l} + mp_l}{m + \textit{In-Topic}_{i,l} + \textit{Out-of-Topic}_{i,l}}. \quad (1)$$

For the particular binary classification task i, $\textit{In-Topic}_{i,l}$ and $\textit{Out-of-Topic}_{i,l}$ are the number *in* and *out* of topic of those training examples that fall in the leaf node, respectively. p_l is the prior *at the leaf node* of *In-Topic* obtained from using all of the data across tasks. m is the effective sample size which determines how much evidential weight, measured in "number of observed datapoints", the prior carries.

We sampled the two extremes of this spectrum, $m = 0$ and $m = \infty$. By choosing $m = 0$, we specialize the *LABEL* model to a particular task by placing all of the weight on the task-specific data. This allows some leaves to effectively have no data in them; for those leaves, we use the overall prior of in-topic according to the task-specific data. We refer to this system as *LABEL-D (repop)* since this acts as if we completely repopulated the decision tree with task-specific data.

We also present the results obtained by making the prediction at a leaf node using all of the data across tasks equally (*i.e.*, $m = \infty$ and the right side of Equation 1 simply becomes p_l). We refer to this as *LABEL-D (general)* since the metaclassifier is not specialized for each task other than the decision threshold. Comparing these specific instantiations allows us to determine if we are simply coincidentally finding a better tree structure using all the data or if the actual predictions based on all the data aids us as well.

4.3. Data

MSN Web Directory The MSN Web Directory is a large collection of heterogeneous web pages (from a May 1999 web snapshot) that have been hierarchically classified. We used the same train/test split of 50078/10024 documents as that reported by Dumais and Chen (2000).

The MSN Web hierarchy is a seven-level hierarchy; we used all 13 of the top-level categories. The class proportions in the training set vary from 1.15% to 22.29%. In the testing set, they range from 1.14% to 21.54%. The classes are general subject categories such as *Health & Fitness* and *Travel & Vacation*. Human indexers have assigned the documents to zero or more categories.

For the experiments below, we used only the top 1000 words with highest mutual information for each class; approximately 195K words appear in at least three training documents.

Reuters

The Reuters 21578 corpus (Lewis, 1997) contains Reuters news articles from 1987. For this data set, we used the ModApte standard train/test split of 9603/3299 documents (8676 unused documents). The classes are economic subjects (*e.g.*, "acq" for acquisitions, "earn" for earnings, etc.)

that human taggers applied to the document; a document may have multiple subjects. Limiting to the ten largest classes allows us to compare our results to a variety of previously published results (Dumais et al., 1998; Joachims, 1998; McCallum & Nigam, 1998; Platt, 1999). The class proportions in the training set vary from 1.88% to 29.96%. In the testing set, they range from 1.7% to 32.95%.

For the experiments below we used only the top 300 words with highest mutual information for each class; approximately 15K words appear in at least three training documents.

TREC-AP

The TREC-AP corpus is a collection of AP news stories from 1988 to 1990. We used the same train/test split of 142791/66992 documents that was used by Lewis et al. (1996). As described by Lewis and Gale (1994) (see also Lewis, 1995), the categories are defined by keywords in a keyword field. The title and body fields are used in the experiments below. There are twenty categories in total. The frequencies of the twenty classes are the same as those reported in Lewis et al. (1996). The class proportions in the training set vary from 0.06% to 2.03%. In the testing set, they range from 0.03% to 4.32%.

For the experiments described below, we used only the top 1000 words with the highest mutual information for each class; approximately 123K words appeared in at least 3 training documents.

4.4. Performance Measures

To compare the performance of the classification methods we look at a set of standard accuracy measures. The F1 measure (van Rijsbergen, 1979; Yang & Liu, 1999) is the harmonic mean of precision and recall where $Precision = \frac{Correct\ Positives}{Predicted\ Positives}$ and $Recall = \frac{Correct\ Positives}{Actual\ Positives}$. Additionally, we report error, emphasizing the *normalized error* score. Normalized error divides the error in each task by the error that would have been achieved by random guessing (the *a priori* prevalent class). A normalized error less than one indicates the method outperforms random guessing. The scores reported here are the arithmetic averages of the values across all tasks (for F1 this is termed macroF1 in the text classification literature).

5. Experimental Results

Table 1 summarizes the performance of the systems over all 43 classification tasks. Better performance is indicated by larger F1 and by smaller error or normalized error values. The best performance in each column is given bold. To determine statistical significance for the macro-averaged measures, a one-sided macro sign test was performed (Yang & Liu, 1999). When comparing system A and system B the null hypothesis is that system A performs better on approximately half the tasks that they differ in performance over. The results for *LABEL-D (general)* are significantly better than the other systems at the $p = 0.01$ level with the exception of the difference between the error metrics of *LABEL-D (general)* and *STRIVE-D* which are significant at the $p = 0.05$ level.

Table 1. Performance Summary over all Tasks

Method	Macro F1	Error	norm Error
Smox	0.7411	0.0197	0.4789
STRIVE-D	0.7457	0.0191	0.4716
LABEL-D (repop)	0.7431	0.0188	0.4758
LABEL-D (general)	**0.7545**	**0.0181**	**0.4512**

6. Discussion and Summary

First, we note that the base classifiers are competitive and more particularly the results for *Smox* are consistent with the best reported results over these corpora (Dumais & Chen, 2000; Dumais et al., 1998; Joachims, 1998). Thus, we are challenged with an extremely competitive baseline.

In spite of this, *LABEL-D (general)* shows dominance for each of the performance measures. Additionally, comparing it directly to the most comparable version of *STRIVE-D* reported in Bennett et al. (2002; 2003), we see improvement over the same system that uses data from each task in isolation. Additionally, by comparing *LABEL-D (general)* to *LABEL-D (repop)*, we see that it is not simply the structure of the resulting decision trees, but that the predicted probabilities induced across the entire set of tasks are key to improving generalization performance. While the percentage improvement is small, we believe these results are very encouraging for the future use of inductive transfer to improve models of classifier reliability.

Similar results are observed for each corpus individually but are more pronounced in the Reuters corpus than the MSN Web or TREC-AP corpora.

7. Future Work

We are currently pursuing several extensions of this work. We are exploring parametric variations of m that control how much weight task-specific data is given versus the weight given to data from across all tasks. Additionally, we are interested in the value of using task identifiers with each example given to the metaclassifier. Such identifiers will allow the metaclassifier to model a task separately if it improves model fit. Finally, we are conducting task-centric analyses of the successes of the approach to improve our understanding of when use of the methods will likely lead to increased predictive performance on a task.

Acknowledgments

We thank Max Chickering for his special support of the WinMine toolkit. Thanks also to Jaime Carbonell and John Lafferty for their useful feedback and suggestions.

References

Bennett, P. N., Dumais, S. T., & Horvitz, E. (2002). Probabilistic combination of text classifiers using reliability indicators: Models and results. *SIGIR '02* (pp. 207–214).

Bennett, P. N., Dumais, S. T., & Horvitz, E. (2003). The combination of text classifiers using reliability indicators. In submission to *Information Retrieval*.

Bollacker, K. D., & Ghosh, J. (1998). A supra-classifier architecture for scalable knowledge reuse. *ICML '98* (pp. 64–72).

Breiman, L., & Friedman, J. H. (1995). *Predicting multivariate responses in multiple linear regression* (Technical Report). ftp://ftp.stat.berkeley.edu/pub/-users/breiman/curds-whey-all.ps.Z.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*, 41–75.

Chickering, D., Heckerman, D., & Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. *UAI '97* (pp. 80–89).

Cohen, W. W., & Kudenko, D. (1997). Transferring and retraining learned information filters. *AAAI '97* (pp. 583–590).

Dietterich, T. (2000). Ensemble methods. *Multiple Classifier Systems, First International Workshop* (pp. 1–15). Springer.

Dumais, S. T., & Chen, H. (2000). Hierarchical classification of web content. *SIGIR '00* (pp. 256–263).

Dumais, S. T., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *CIKM '98* (pp. 148–155).

Gama, J. (1998). Combining classifiers by constructive induction. *ECML '98* (pp. 178–189).

Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R., & Kadie, C. (2000). Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, *1*, 49–75.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *ECML '98* (pp. 137–142).

Lewis, D. D. (1995). A sequential algorithm for training text classifiers: Corrigendum and additional data. *SIGIR Forum*, *29*, 13–19.

Lewis, D. D. (1997). Reuters-21578, distribution 1.0. http://www.daviddlewis.com/resources/ testcollections/reuters21578.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. *SIGIR '94* (pp. 3–12).

Lewis, D. D., Schapire, R. E., Callan, J. P., & Papka, R. (1996). Training algorithms for linear text classifiers. *SIGIR '96* (pp. 298–306).

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. *AAAI '98, Workshop on Learning for Text Categorization*.

Microsoft Corporation (2001). WinMine Toolkit v1.0. http://research.microsoft.com/˜dmax/WinMine/-ContactInfo.html.

Platt, J. C. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. Burges and A. J. Smola (Eds.), *Advances in kernel methods – support vector learning*. MIT Press.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. J. Smola, P. Bartlett, B. Scholkopf and D. Schuurmans (Eds.), *Advances in large margin classifiers*. MIT Press.

Thrun, S., & O'Sullivan, J. (1996). Discovering structure in multiple learning tasks: The tc algorithm. *ICML '96* (pp. 489–497).

Ting, K., & Witten, I. (1999). Issues in stacked generalization. *Journal of Artificial Intelligence Research*, *10*, 271–289.

Toyama, K., & Horvitz, E. (2000). Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking. *ACCV 2000, Fourth Asian Conference on Computer Vision*.

van Rijsbergen, C. (1979). *Information retrieval*. Butterworths, London.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *SIGIR '99* (pp. 42–49).