Submitted (1/98) to the *15th National Conference on Artificial Intelligence (AAAI-98).*

Learning to Classify Text from Labeled and Unlabeled Documents

Kamal Nigam[†] knigam@cs.cmu.edu
Andrew McCallum[‡†] mccallum@cs.cmu.edu
Sebastian Thrun[†] thrun@cs.cmu.edu
Tom Mitchell[†] mitchell+@cs.cmu.edu

[†]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[‡]Justsystem Pittsburgh Research Center
4616 Henry Street
Pittsburgh, PA 15213

Abstract

This paper shows that the accuracy of learned text classifiers can be improved by augmenting small numbers of labeled training documents with a large pool of unlabeled documents. This is significant because in many important text classification problems obtaining classification labels is expensive, while large quantities of unlabeled documents are readily available. We present a theoretical argument showing that, under common assumptions, unlabeled data contain information about the target function. We then introduce an algorithm for learning from labeled and unlabeled text, based on combining Expectation-Maximization with a naive Bayes classifier. The algorithm first trains a classifier using the available labeled documents, and probabilistically labels the unlabeled documents. It then trains a new classifier using the labels for all the documents, and iterates. Experimental results, obtained using text from three different real-world tasks, show that the use of unlabeled data reduces classification error up to 30%.

Introduction

Consider the problem of training a computer to automatically classify text documents. Given the growing volume of online text available through the World Wide Web, Internet news feeds, electronic mail, and digital libraries, this problem is of great practical significance. There are statistical text learning algorithms that can be trained to approximately classify documents, given a sufficient set of labeled training examples. These text classification algorithms have been used to automatically catalog news articles (Lewis & Gale 1994) and web pages (Craven *et al.* 1998), automatically learn the reading interests of users (Pazzani, Muramatsu, & Billsus 1996; Lang 1995), and automatically sort electronic mail (Lewis & Knowles 1997).

One key difficulty with these current algorithms, and the issue addressed by this paper, is that they require a large, often prohibitive, number of labeled training examples to learn accurately. Take, for example, the task of learning which newsgroup articles are of interest to a person reading UseNet news, as examined by Lang (Lang 1995). After reading and classifying about 1000 articles, precision of the learned classifier was about 50% for the top 10% of documents ranked by the classifier. As a practical user of such a filtering system, one would obviously prefer learning algorithms that can provide accurate classifications after hand-labeling only a few dozen articles, rather than thousands.

In this paper we describe an algorithm that learns to classify text documents more accurately by using *unlabeled* documents to augment the available *labeled* training examples. In our example, the labeled documents might be just 10 articles that have been read and judged by the user as interesting or not. Our learning algorithm can make use of the vast multitude of unlabeled articles available on UseNet to augment these 10 labeled examples. In many text domains, especially those involving online sources, collecting unlabeled examples is trivial; it is the labeling that is expensive.

We present experimental results showing that this unlabeled data can boost learning accuracy in three domains: classifying newsgroup articles, web pages, and Reuters news articles. For example, to identify the source newsgroup for a UseNet article with 70% classification accuracy, our algorithm takes advantage of 10,000 unlabeled examples and requires only 300 labeled examples, while a traditional learner requires 1000 labeled examples to achieve the same accuracy. The technique thus effectively triples the number of labeled training examples in this instance.

Why do unlabeled examples boost learning accuracy? In brief, they provide information about the joint probability distribution over words within the documents. Suppose, for example, that using only the labeled data we determine that documents containing the word "learn" tend to belong to the positive class. If we use this fact to estimate the classification of the many unlabeled documents, we might find that the word "teach" occurs frequently in the unlabeled examples that are now believed to belong to the positive class. Thus the co-occurrence of the words "learn" and "teach" over the large set of unlabeled training data can provide useful information to con-

struct a more accurate classifier that considers both "learn" and "teach" as indicators of positive examples.

The specific approach we describe here is based on a combination of two well-known learning algorithms: the naive Bayes classifier (Lewis & Ringuette 1994) and the Expectation Maximization (EM) algorithm (Dempster, Laird, & Rubin 1977). The naive Bayes algorithm is a one of a class of statistical text classifiers that use word frequencies as features. Other examples include Rocchio (Salton 1991; Rocchio 1971), regression models (Yang & Chute 1993), k-nearest-neighbor (Yang & Pederson 1997) and Support Vector Machines (Joachims 1997b). EM is a class of iterative algorithms for maximum likelihood estimation in problems with incomplete data. The result is an algorithm that extends conventional text learning algorithms by using EM to dynamically derive pseudo-labels for unlabeled documents during learning, thereby providing a way for incorporating unlabeled data into supervised learning. Previous supervised algorithms for learning to classify from text do not incorporate unlabeled data.

A similar approach was used by Miller and Uyar (Miller & Uyar 1997) for non-text data sources. We adapt this approach for the naive Bayes text classifier and conduct a thorough empirical analysis. We also present a formal result, showing that for a much larger class of learning algorithms—which includes many of the text learning algorithms mentioned above—unlabeled data indeed carries information about the target function.

In the following sections, we provide sufficient conditions under which the use of unlabeled data can be expected to improve classification accuracy. We present the probabilistic setting for naive Bayes, and its incorporation with the Expectation Maximization algorithm. We empirically demonstrate significantly improved performance on three text data sets. Finally, we discuss directions for future research and argue the practical importance of these results, since in many text learning domains (such as the Web), unlabeled data is available almost for free, whereas labeling the data can be truly expensive.

The Probabilistic Framework

To ground the theoretical aspects of our work, and to provide a setting for our algorithm, this section presents a probabilistic framework for characterizing the nature of documents and classifiers. We will then use this to introduce the classifier and show that unlabeled data can be used to improve classification. The framework follows commonly used assumptions (Lewis & Ringuette 1994; Domingos & Pazzani 1997) about the data—that our text is produced by a mixture model, and that there is a one-to-one correspondence between mixture components and classes.

The data set is represented as a collection of documents, $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$. Every document is generated by a probability distribution given by a mixture model parameterized by θ. The mixture model contains $j = \{1, \ldots, |\mathcal{C}|\}$ mixture components, each parameterized by θ_j, which are disjoint subsets of θ. A document d_i is produced by first selecting a mixture component according to a probability distribution given by $\mathrm{P}(c_j|\theta) = \theta_{0j}$. The selected component then produces a document according to its own parameterization, θ_j. We can characterize the likelihood of a document with a sum of total probability over all mixture components:

$$\mathrm{P}(d_i|\theta) = \sum_{j=1}^{|C|} \mathrm{P}(d_i|c_j;\theta_j)\mathrm{P}(c_j|\theta) \qquad (1)$$

Each document has a class label chosen from $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$, where the class label of document d_i is written $l_i \in \mathcal{C}$. We assume that there is a one-to-one correspondence between class c_j and mixture component j, and refer to the *class label* of a document as the index of the mixture component that generated the document. This class label may or may not be known for a given document.

Proof of the Value of Unlabeled Data

In this section we will show that, in this setting, documents with unknown class labels are useful for learning concept classes. 'Learning concept classes' in this scenario is equivalent to estimating parameters of an unknown mixture model that produced the given training documents. We assume that the learning task is not *degenerate*, that is,

$$\begin{aligned} \exists d_i, c_j, \theta, \theta'. \quad \mathrm{P}(d_i|c_j;\theta_j) &\neq \mathrm{P}(d_i|c_j;\theta'_j) \\ \wedge\ \mathrm{P}(c_j|\theta) &\neq \mathrm{P}(c_j|\theta'). \end{aligned} \qquad (2)$$

This assumption excludes tasks where learning is impossible, even given training data with class labels, for the reason that all hypotheses for θ yield equivalent results. In this case knowledge of the parameters would not help aid the prediction of future data.

To show that knowledge about an unlabeled document d_i carries information about the parameters θ, we need to demonstrate the conditional dependence of θ on d_i. That is:

$$\mathrm{P}(\theta|d_i) \neq \mathrm{P}(\theta). \qquad (3)$$

If this conjecture holds, a direct implication is that unlabeled data convey information about the parameters of θ. In this way, unlabeled data could help in supervised learning.

We provide a proof by contradiction. For this, we temporarily assume that θ and d_i are independent.

By negating Equation 3, and applying Bayes' rule, we write:

$$\forall\theta.\ \mathrm{P}(d_i|\theta) = \mathrm{P}(d_i). \tag{4}$$

One direct conclusion of this equation is that any two parameterizations, θ and θ', provide the same document probabilities for any document. By substitution from Equation 1 this gives:

$$\sum_{j=1}^{|C|} \mathrm{P}(d_i|c_j;\theta_j)\mathrm{P}(c_j|\theta) = \sum_{j=1}^{|C|} \mathrm{P}(d_i|c_j;\theta'_j)\mathrm{P}(c_j|\theta'). \tag{5}$$

From here, it is a straightforward exercise to construct a document and two parameterizations to generate a contradiction. Our assumption of non-degeneracy requires that for some document the individual terms in Equation 5 must differ for some θ and θ'; we can construct one for which the total probability of the document also differs for θ and θ'. Thus, our assumption of conditional independence is contradicted, and parameterizations must be conditionally dependent on the documents. This signifies that unlabeled data indeed convey information about parameters of the generative model.

Naive Bayes for Text Classification

In this section, we present the naive Bayes classifier—a well-known, probabilistically based algorithm for classifying text; furthermore, naive Bayes is a special case of the mixture model assumption. The fundamentals of this algorithm will later be used in incorporating unlabeled data. We build upon the previous section, and continue the use of the two assumptions for data generation: that the documents are produced according to a mixture model, and that there is a one-to-one correspondence between classes and mixture components. Thus, the motivational result from the previous section still holds—that unlabeled documents can be beneficial. Naive Bayes uses the additional assumption that the probability of seeing a word in a document is independent of its context and its position. (Lewis & Ringuette 1994; Domingos & Pazzani 1997)

The learning task is to take a set of training documents and form estimates for the parameters of the generative model. Naive Bayes forms Bayes optimal estimates of these parameters, then uses the estimated model to classify new documents.

Document d_i is considered an ordered list of word events d_{ik}, each of which corresponds to seeing some word in position k of document d_i. Given this, we can expand the first term from Equation 1:

$$\mathrm{P}(d_i|c_j;\theta_j) = \prod_{k=1}^{|d_i|} \mathrm{P}(d_{ik}|c_j;\theta_j;d_{iq}, q<k). \tag{6}$$

Next we make the naive Bayes assumption: that the words of a document are generated independently of their context. We further assume the probability of the word is independent of its position within the document. Thus, each document d_i is drawn from a multinomial distribution with as many independent trials as the number of words in d_i. We write the nth word in the vocabulary V as w_n; then given $d_{ik} = w_n$, express the assumption as:

$$\mathrm{P}(d_{ik}|c_j;\theta_j;d_{iq}, q<k) = \mathrm{P}(w_n|c_j;\theta_j). \tag{7}$$

The parameters of a mixture component, θ_j, are exactly these word probabilities, such that $\theta_{jn} = \mathrm{P}(w_n|c_j;\theta_j)$, where $n = \{1,\ldots,|V|\}$.

Given these underlying assumptions of how the data is produced, the task of learning a text classifier consists of forming an estimate for θ given a set of data and their associated class labels. The Bayes optimal estimates for the word probabilities θ_{jn} are (Vapnik 1982)

$$\theta_{jn} = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_n,d_i)\mathrm{P}(c_j|d_i)}{|V| + \sum_{m=1}^{|V|}\sum_{i=1}^{|\mathcal{D}|} N(w_m,d_i)\mathrm{P}(c_j|d_i)}. \tag{8}$$

where $N(w_n,d_i)$ is the count of the number of times word w_n occurs in document d_i and where $\mathrm{P}(c_j|d_i) = \{0,1\}$ given by the class label. This follows directly from the multinomial event model with the addition of Bayes optimal smoothing to prevent zero probabilities. The class prior probabilities, θ_{0j} are estimated by

$$\theta_{0j} = \frac{\sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(c_j|d_i)}{|\mathcal{D}|}. \tag{9}$$

To apply a θ estimated in this way to the task of text classification, we calculate the probabilities of class labels given a new, unlabeled document, d_i.

$$\begin{aligned}\mathrm{P}(c_j|d_i;\theta) &= \frac{\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta_j)}{\mathrm{P}(d_i|\theta)} \\ &= \frac{\mathrm{P}(c_j|\theta)\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_j;\theta_j)}{\sum_{r=1}^{|C|}\mathrm{P}(c_r|\theta)\prod_{k=1}^{|d_i|}\mathrm{P}(w_{d_{ik}}|c_r;\theta_r)}\end{aligned} \tag{10}$$

The first step is an application of Bayes' rule, and the second is a substitution using Equations 1, 6 and 7. If the task is to classify a test document d_i into a single class, simply select $\arg\max_j \mathrm{P}(c_j|d_i;\theta)$.

Note that our three assumptions about the generation of text documents (mixture model, one-to-one correspondence between mixture components and

classes, and word independence) are violated in practice. Documents often fall into overlapping categories. Words within a document are not independent of each other—grammar and topicality ensure this. Despite these violations, empirically, the Naive Bayes classifier does a good job of classifying text documents (Lewis & Ringuette 1994; Craven *et al.* 1998; Yang & Pederson 1997; Joachims 1997a). Domingos and Pazzani shed some light on why the word independence assumption can be so violated, yet performance remains so good (Domingos & Pazzani 1997).

Using EM to Incorporate Unlabeled Data

Now we address the question of how to combine labeled and unlabeled training documents to produce a good estimate of θ. This is a special case of the more general missing values formulation, as presented by (Ghahramani & Jordan 1994).

Consider the probability of all training data. Some documents $d_i \in \mathcal{D}^l$ come with class labels $l_i \in \mathcal{C}$; for other documents $\mathcal{D}^u$, the class labels are unknown. This forms a disjoint partitioning of $\mathcal{D}$, such that $\mathcal{D} = \mathcal{D}^l \cup \mathcal{D}^u$. From Equation 1, and assuming all documents were generated independently of each other, the probability of all the data is

$$\begin{aligned} \mathrm{P}(\mathcal{D}|\theta) &= \prod_{i=1}^{|\mathcal{D}^u|} \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(d_i|c_j;\theta_j)\mathrm{P}(c_j|\theta) \\ &\quad \times \prod_{i=1}^{|\mathcal{D}^l|} \mathrm{P}(d_i|c_j = l_i;\theta_j)\mathrm{P}(c_j = l_i|\theta) \end{aligned} \tag{11}$$

Again, learning a classifier corresponds to calculating a maximum likelihood estimate of θ. By Bayes' rule, $\mathrm{P}(\mathcal{D}|\theta) = \mathrm{P}(\theta|\mathcal{D})\mathrm{P}(\mathcal{D})/\mathrm{P}(\theta)$. $\mathrm{P}(\mathcal{D})$ is a constant; maximum likelihood estimation assumes that $\mathrm{P}(\theta)$ is a constant, so taking the log, we say $\eta = \log(\mathrm{P}(\mathcal{D})/\mathrm{P}(\theta))$, and get the log likelihood of the parameters given the data, $l(\theta|\mathcal{D})$:

$$\begin{aligned} l(\theta|\mathcal{D}) &= \eta + \sum_{i=1}^{|\mathcal{D}^u|} \log \sum_{j=1}^{|\mathcal{C}|} \mathrm{P}(d_i|c_j;\theta_j)\mathrm{P}(c_j|\theta) \\ &\quad + \sum_{i=1}^{|\mathcal{D}^l|} \log\left(\mathrm{P}(d_i|c_j = l_i;\theta_j)\mathrm{P}(c_j = l_i|\theta)\right) \end{aligned} \tag{12}$$

Because the first line of the equation has a log of sums, it is not computable in closed-form. However, if we knew all the class labels, as in $\mathcal{D}^l$, then we could avoid this. If we had class labels $\mathbf{z}$, $\mathbf{z}_i = \langle z_{i1}, \ldots, z_{i|\mathcal{C}|}\rangle$, where $z_{ij} = 1$ iff $l_i = c_j$ else $z_{ij} = 0$, then we have the following closed-form solution for the complete log likelihood, $l_c(\theta|\mathcal{D}, \mathbf{z})$

- Build an initial classifier by estimating θ_{0j} and θ_j from the labeled documents only (Equations 8 and 9).
- Loop while classifier parameters change:
  - Use the current classifier to probabilistically label the unlabeled documents (Equation 10).
  - Recalculate the classifier parameters θ_{0j} and θ_j given the probabilistically assigned labels (Equations 8 and 9).

Table 1: The Algorithm

$$l_c(\theta|\mathcal{D}, \mathbf{z}) = \eta + \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(d_i|c_j;\theta_j)\mathrm{P}(c_j|\theta)\right) \tag{13}$$

Note that $\mathbf{z}_i$ is already known for labeled documents. Dempster (Dempster, Laird, & Rubin 1977) shows that by working with the expected value of $\mathbf{z}$ at iteration k, denoted by $Q^{(k)}$, we can find a local maximum for $l(\theta|\mathcal{D})$ by iterating the following two steps:

- E-step: Set $Q^{(k)} = E[\mathbf{z}|\mathcal{D};\theta^{(k)}]$.
- M-step: Set $\theta^{(k+1)} = \arg\max_\theta \mathrm{P}(\theta|\mathcal{D};Q^{(k)})$.

In practice, the E-step corresponds to calculating $\mathrm{P}(c_j|d_i;\theta)$ for every document by using the current estimate of θ and Equation 10. The M-step corresponds to calculating a new maximum likelihood estimate for θ given the current estimates for $\mathrm{P}(c_j|d_i;\theta)$ using Equations 8 and 9. See Table 1 for an outline of our algorithm.

EM finds the θ that locally maximizes the probability of all the data, both the labeled and the unlabeled.

Experimental Results

In this section, we give empirical evidence that using the algorithm in Table 1 with labeled and unlabeled documents outperforms naive Bayes, which does not on its own use unlabeled documents. We present experimental results with three different text corpora.[1] In two cases we restrict ourselves to the more populous classes of a data set in order to have enough unlabeled data to perform meaningful experiments.

The 20 Newsgroups data set (Joachims 1997a), collected by Ken Lang, consists of 20,017 articles divided almost evenly among 20 different UseNet discussion

[1] All three of these data sets are available on the Internet. See http://www.cs.cmu.edu/~textlearning and http://www.research.att.com/~lewis.

groups. When words from a stoplist of common short words are removed, there are 62,258 unique words that occur more than once. Many of the categories fall into confusable clusters; for example, four of them are `comp.*` discussion groups, and three of them discuss religion. When tokenizing this data, we skip the UseNet headers (thereby discarding the subject line); tokens are formed from contiguous alphabetic characters, which are left unstemmed. Best performance was obtained with no feature selection, and by normalizing word counts by document length. On each trial, 20% of the documents were randomly selected for placement in the test set.

The WebKB data set (Craven *et al.* 1998) contains 8145 web pages gathered from university computer science departments. For four departments, all web pages were included; additionally, there are many pages from an assortment of other universities. The pages are divided into seven categories: student, faculty, staff, course, project, department and other. In this paper, we use the four most populous non-other categories: student, faculty, course and project, all together containing 4199 pages. We did not use stemming or a stoplist; we found that using a stoplist actually hurt performance because, for example, "my" is the fourth-ranked word by information gain, and is an excellent indicator of a student homepage. As done previously (Craven *et al.* 1998), we use only the 2000 most informative words, as measured by information gain. Accuracy results presented below are an average of four test/train splits, holding out and testing on each complete university in turn.

The 'ModApte' train/test split of the Reuters21578 Distribution 1.0 data set consists of 12902 articles and 90 topic categories from the Reuters newswire. Following several other studies (Joachims 1997b; Liere & Tadepalli 1997) we use the 10 most populous classes and build binary classifiers for each class. We use a stoplist, but do not stem. Results are reported on the complete test set as precision-recall breakeven points, a standard information retrieval measure for binary classification.

In experiments with EM, the initial θ was estimated using only the labeled data, and EM iterations progressed from there, as in Table 1. All experiments were performed with eight EM iterations; significant changes occur in the first few iterations. We never found classification accuracy to improve beyond the eighth iteration.

Figure 1 shows the effect of using EM with unlabeled data in the 20 Newsgroups data set. In the top graph, the vertical axis indicates classifier accuracy on test sets, and the horizontal axis indicates the amount of labeled data used in training. We vary the amount of labeled training data, and compare the classification accuracy of traditional naive Bayes (no unlabeled

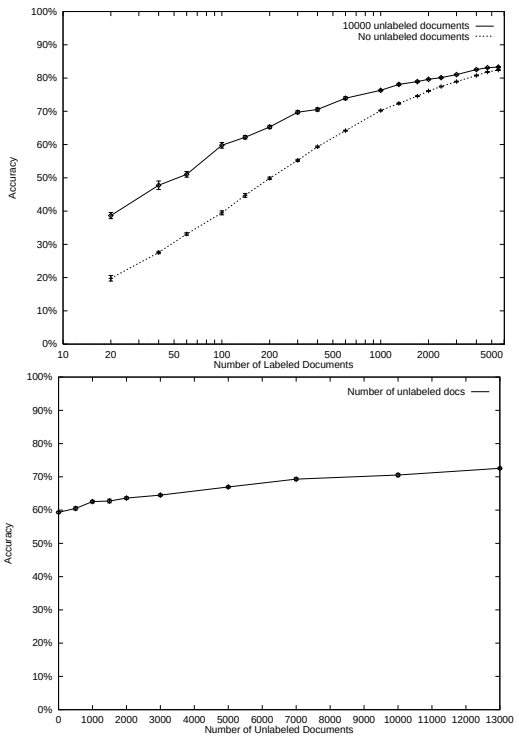

Figure 1: **Top:** Classification accuracy on the 20 Newsgroups data set, both with and without 10000 unlabeled documents, averaged over 10 trials. The narrow error bars on each data point show standard error. **Bottom:** Classification accuracy with 400 labeled documents, averaged over 10 trials.

data) with an EM learner that has access to 10000 unlabeled documents. EM performs significantly better. For example, with 400 labeled documents (20 documents per class), naive Bayes reaches 59% accuracy, while EM achieves 70%. Note here that EM performs well even with a very small number of labeled documents; with only 20 documents (a single labeled document per class), naive Bayes gets 19%, EM 39%. As expected, when there is a lot of labeled data, and the naive Bayes learning curve has reached a plateau, the learner is already saturated, and having unlabeled data does not help. In the bottom graph, we hold the number of labeled documents constant at 400, and vary the number of unlabeled documents in the horizontal axis. Naturally, more unlabeled data helps.

Table 2 provides a window into the evolution of the classifier over the course of EM iterations for a specific example. Based on the WebKB data set, each column shows the ordered list of words that the model believes are most predictive of the course class. At Iteration 0, the parameters were estimated from a randomly-chosen single labeled document. Notice that the document seems to be about a specific Artificial Intelligence course at Dartmouth. After two EM iterations with 3500 unlabeled documents, we see that EM has used

Iteration 0	Iteration 1	Iteration 2
intelligence	*DD*	*D*
DD	*D*	*DD*
artificial	lecture	lecture
understanding	cc	cc
*DD*w	*D*⋆	*DD:DD*
dist	*DD:DD*	due
identical	handout	*D*⋆
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	*DD*am	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	*DD*am
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

Table 2: Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific example. The symbol *D* indicates an arbitrary digit.

the unlabeled data to find words that are more generally indicative of courses. The classifier corresponding to the first column gets 50% accuracy; by the eighth iteration, the classifier achieves 71% accuracy.

The graph in Figure 2 shows the benefits of 3500 unlabeled documents on the WebKB data set. Again, EM improves accuracy significantly, especially when the amount of labeled data is small. When there is a lot of labeled data however, EM hurts performance slightly. We hypothesize that the reason for this is that the data does not fit the assumptions of our model as well as 20 Newsgroups—that is, the mixture components that best explain the unlabeled data do not correspond as well to the class labels.

With an eye towards testing this hypothesis, we applied EM to the Reuters21578 corpus. Since the documents in this data set can have multiple class labels, each category is traditionally evaluated with a binary classifier. Thus, the negative class covers 89 categories, and we expected this task to strongly violate the assumption that all the data for each of the binary classes is generated by a single mixture component. With 40 labeled documents (20 positive and 20 negative) and no unlabeled documents, naive Bayes performed best with no feature selection. The left column of Table 3 shows average precision-recall breakeven points for 10 trials of each experiment. When we applied EM with 7000 unlabeled documents, we found that performance was

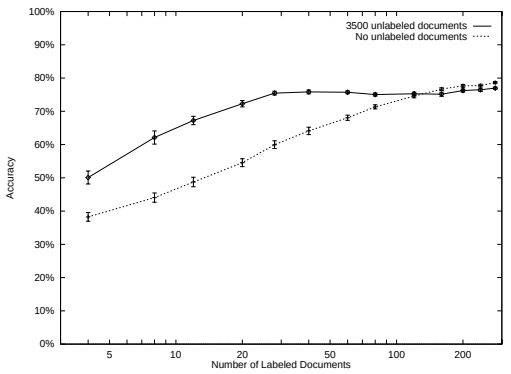

Figure 2: Classification accuracy on the WebKB data set, both with and without 3500 unlabeled documents, averaged over 80 trials per data point.

	N.B.	EM 2	EM 21
acq	69.3	31.2	79.5
corn	39.1	28.4	42.0
crude	50.6	24.3	69.4
earn	91.9	88.6	88.1
grain	47.2	23.6	61.8
interest	47.3	22.9	48.9
money-fx	45.0	29.7	59.7
ship	54.5	8.9	53.6
trade	52.5	31.5	53.9
wheat	45.9	19.2	45.5

Table 3: Precision-Recall breakeven points showing performance of binary classifiers on Reuters with traditional naive Bayes, EM with one mixture component per class, and EM with 20 mixture components on the negative class.

best by selecting only the top 200 words by information gain. As shown in the middle column of Table 3, using the unlabeled data does indeed result in dramatically worse performance. Next, we changed our model in a way that allows the natural clusters in the data to be in better correspondence with the class labels—we used 20 mixture components for the negative class instead of one. The right column shows that in seven out of ten cases, using EM improves performance over traditional naive Bayes. A paired T-test on each trial over all categories shows that the improvement in average breakeven point from 57% with naive Bayes to 60% with 20-component EM is statistically significant ($p < 0.05$). In future work we intend to apply this multi-component approach to more problems.

Related Work

Two other studies have used EM to combine labeled and unlabeled data for classification (Miller & Uyar 1997; Shahshahani & Landgrebe 1994). Instead of naive Bayes, Shahshahani and Landgrebe use a mix-

ture of Gaussians; Miller and Uyar use Mixtures of Experts. They demonstrate experimental results on non-text data sets with up to 40 features. In contrast, our textual data sets have three orders of magnitude more features. Shahshahani and Landgrebe present a proof that unlabeled data reduces variance in parameter estimation. Their proof does not apply in our case, however, because our target concept can not be learned without some labeled data. (There is no efficient estimator for *all* parameters.)

Our work is an example of applying EM to fill in missing values—the missing values are the class labels of the unlabeled training examples. Ghahramani and Jordan have used EM with mixture models to fill in missing values (Ghahramani & Jordan 1994). The emphasis of their work was on missing feature values, where we focus on augmenting a very small but complete set of labeled data.

The AutoClass project (Cheeseman & Stutz 1996) investigated the combination of the EM algorithm with an underlying model of a Naive Bayes classifier. The emphasis of their research was the discovery of novel clusterings for unsupervised learning over unlabeled data. AutoClass has not been applied to text or classification.

Many approaches to reducing the need for labeled training examples have used active learning, in which an algorithm iteratively selects an unlabeled example, asks a human labeler for its classification, and rebuilds its classifier. Approaches differ in their methods for selecting the unlabeled example to classify. Three such examples are relevance sampling (Salton & Buckley 1990), uncertainty sampling (Lewis & Gale 1994), and a "Query By Committee" approach (Liere & Tadepalli 1997).

On several corpora, some other statistical text classifiers (Yang & Pederson 1997; Joachims 1997b; Cohen & Singer 1997) have been shown to mildly outperform naive Bayes in text classification. However, naive Bayes has a strong probabilistic foundation for EM, and is more efficient for large data sets. The thrust of this paper is to straightforwardly demonstrate the value of unlabeled data; future work will apply unlabeled data to more complex classifiers.

Summary and Conclusions

This paper has explored the question of when and how unlabeled data may be used to supplement scarce labeled data in machine learning problems, especially when learning to classify text documents. This is an important question in text learning, because of the high cost of hand-labeling data and because of the availability of huge volumes of unlabeled data. In this paper we have presented a theoretical model, an algorithm, and experimental results that show significant improvements from using unlabeled documents for training classifiers in three real-world text classification tasks.

Our theoretical model characterizes a setting in which unlabeled data can be used to boost the accuracy of learned classifiers: when the probability distribution that generates documents can be described as a mixture distribution, and where the mixture components correspond to the class labels. These conditions fit exactly the model used by the naive Bayes classifier.

However, the complexity of natural language text will not soon be completely captured by statistical models. It is interesting then, to consider the sensitivity of a classifier's model to data inconsistent with that model. Domingos and Pazzani analyze this for the word independence assumption (Domingos & Pazzani 1997). With our results on Reuters21578 we begin to study cases in which the data violate the assumptions about mixture models and the correspondence between components and classes. The results suggest exploring the use of more complex mixture models that better correspond to textual data distributions.

We believe that our algorithm and others using unlabeled data require a closer match between the data and the model than those with only labeled data; if the intended target concept and model contrast too much with the natural distribution of the data, then the use of unlabeled data will hurt instead of help. We intend to make a closer theoretical and empirical study on the tradeoffs between the use of unlabeled data and the inherent model inadequacies for several text learning algorithms.

We also see several other interesting directions for future work with unlabeled data. Two other learning task formulations could also benefit from using EM: (1) an active learning approach that uses an explicit model of unlabeled data could incorporate EM iterations at every stage to improve its classification, and to better select for which data to request class labels; (2) an incremental learning algorithm that retrains throughout the testing phase could use the unlabeled test data received early in the testing phase in order to improve performance on the later test data.

Other problem domains share some similarities with text domains, and also have abundant unlabeled data with limited, expensive labeled data. Robotics, vision, and information extraction are three such domains. Applying the techniques in this paper could improve classification in these areas as well.

Acknowledgments

We thank Doug Baker for help formatting the Reuters data set. This research was supported in part by the Darpa HPKB program under contract F30602-97-1-0215.

References

Cheeseman, P., and Stutz, J. 1996. Bayesian classification (AutoClass): Theory and results. In Fayyad, U., ed., *Advances in Knowledge Discovery and Data Mining*.

Cohen, W., and Singer, Y. 1997. Context-sensitive learning methods for text categorization. In *Proceedings of ACM SIGIR Conference*.

Craven, M.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Quek, C. 1998. Learning to extract symbolic knowledge from the World Wide Web. Technical report, School of Computer Science, CMU.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM. algorithm. *Journal of the Royal Statistical Society, Series B* 39:1–38.

Domingos, P., and Pazzani, M. 1997. Beyond independence: Conditions for the optimality of the simple bayesian classifier. *Machine Learning* 29:103–130.

Ghahramani, Z., and Jordan, M. 1994. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems (NIPS 6)*. Morgan Kauffman Publishers.

Joachims, T. 1997a. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *International Conference on Machine Learning (ICML)*.

Joachims, T. 1997b. Text categorization with Support Vector Machines: Learning with many relevant features. Technical Report LS8-Report, University of Dortmund.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *International Conference on Machine Learning (ICML)*, 331–339.

Lewis, D., and Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of ACM SIGIR Conference*.

Lewis, D. D., and Knowles, K. A. 1997. Threading electronic mail: A preliminary study. *Information Processing and Management* 33(2):209–217.

Lewis, D., and Ringuette, M. 1994. A comparison of two learning algorithms for text categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, 81–93.

Liere, and Tadepalli. 1997. Active learning with committees for text categorization. In *AAAI-97*.

Miller, D. J., and Uyar, H. S. 1997. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Advances in Neural Information Processing Systems (NIPS 9)*.

Pazzani, M. J.; Muramatsu, J.; and Billsus, D. 1996. Syskill & Webert: Identifying interesting Web sites. In *AAAI-96*.

Rocchio, J. 1971. Relevance feedback in information retrieval. In *The SMART Retrieval System:Experiments in Automatic Document Processing*. Prentice Hall. chapter 14, 313–323.

Salton, G., and Buckley, C. 1990. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* 41(4):288–297.

Salton, G. 1991. Developments in automatic text retrieval. *Science* 253:974–979.

Shahshahani, B., and Landgrebe, D. 1994. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. on Geoscience and Remote Sensing* 32(5):1087–1095.

Vapnik, V. 1982. *Estimations of dependences based on statistical data*. Springer Publisher.

Yang, Y., and Chute, C. G. 1993. An application of least squares fit mapping to text information retrieval. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference*.

Yang, Y., and Pederson, J. 1997. Feature selection in statistical learning of text categorization. In *ICML-97*, 412–420.