Using Maximum Entropy for Text Classification

Kamal Nigam†
knigam@cs.cmu.edu

John Lafferty†
lafferty@cs.cmu.edu

Andrew McCallum‡†
mccallum@justresearch.com

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

‡Just Research
4616 Henry Street
Pittsburgh, PA 15213

Abstract

This paper proposes the use of maximum entropy techniques for text classification. Maximum entropy is a probability distribution estimation technique widely used for a variety of natural language tasks, such as language modeling, part-of-speech tagging, and text segmentation. The underlying principle of maximum entropy is that without external knowledge, one should prefer distributions that are uniform. Constraints on the distribution, derived from labeled training data, inform the technique where to be minimally non-uniform. The maximum entropy formulation has a unique solution which can be found by the improved iterative scaling algorithm. In this paper, maximum entropy is used for text classification by estimating the conditional distribution of the class variable given the document. In experiments on several text datasets we compare accuracy to naive Bayes and show that maximum entropy is sometimes significantly better, but also sometimes worse. Much future work remains, but the results indicate that maximum entropy is a promising technique for text classification.

1 Introduction

A variety of techniques for supervised learning algorithms have demonstrated reasonable performance for text classification; a non-exhaustive list includes naive Bayes [Lewis, 1998; McCallum and Nigam, 1998; Sahami, 1996], k-nearest neighbor [Yang, 1999], support vector machines [Joachims, 1998; Dumais *et al.*, 1998], boosting [Schapire and Singer, 1999] and rule learning algorithms [Cohen and Singer, 1996; Slattery and Craven, 1998]. Among these, however, no single technique has proven to consistently outperform the others across many domains.

This paper explores the use of maximum entropy for text classification as an alternative to previously used text classification algorithms. Maximum entropy has already been widely used for a variety of natural language tasks, including language modeling [Chen and Rosenfeld, 1999; Rosenfeld, 1994], text segmentation [Beeferman *et al.*, 1999], part-of-speech tagging [Ratnaparkhi, 1996], and prepositional phrase attachment [Ratnaparkhi *et al.*, 1994]. Maximum entropy has been shown to be a viable and competitive algorithm in these domains.

Maximum entropy is a general technique for estimating probability distributions from data. The over-riding principle in maximum entropy is that when nothing is known, the distribution should be as uniform as possible, that is, have maximal entropy. Labeled training data is used to derive a set of constraints for the model that characterize the class-specific expectations for the distribution. Constraints are represented as expected values of "features," any real-valued function of an example. The improved iterative scaling algorithm finds the maximum entropy distribution that is consistent with the given constraints.

In our text classification scenario, maximum entropy estimates the conditional distribution of the class label given a document. A document is represented by a set of word count features. The labeled training data is used to estimate the expected value of these word counts on a class-by-class basis. Improved iterative scaling finds a text classifier of an exponential form that is consistent with the constraints from the labeled data.

Our experimental results show that maximum entropy is a technique that warrants further investigation for text classification. On one data set, for example, maximum entropy reduces classification error by more than 40% compared to naive Bayes. On other data sets, basic maximum entropy does not perform as well as naive Bayes. Here, there is evidence that basic maximum entropy suffers from overfitting and poor feature selection. When a prior is applied to maximum entropy, performance is improved in these cases. Overall, maximum entropy performs better than naive Bayes on two of three data sets. Many areas for further investigation exist which may improve performance even further. These include more appropriate feature selection, using bigrams and phrases as features, and adjusting the prior based on the sparsity of the data.

This paper proceeds as follows. Section 2 presents the general framework for maximum entropy for estimating conditional distributions. Then, the specific application of maximum entropy to text classification is discussed in Section 3. Related work is presented in Section 4. Experimental results are presented in Section 5. Finally, Section 6 discusses plans for future work.

2 Maximum Entropy

The motivating idea behind maximum entropy is that one should prefer the most uniform models that also satisfy any given constraints. For example, consider a four-way text classification task where we are told only that on average 40% of documents with the word "professor" in them are in the faculty class. Intuitively, when given a document with "professor" in it, we would say it has a 40% chance of being a faculty document, and a 20% chance for each of the other three classes. If a document does not have "professor" we would guess the uniform class distribution, 25% each. This model is exactly the maximum entropy model that conforms to our known constraint. Calculating the model is easy in this example, but when there are many constraints to satisfy, rigorous techniques are needed to find the optimal solution. Csiszár [1996] provides a good tutorial introduction to maximum entropy techniques.

In its most general formulation, maximum entropy can be used to estimate any probability distribution. In this paper we are interested in classification; thus we limit our further discussion to learning conditional distributions from labeled training data. Specifically, we learn the conditional distribution of the class label given a document.

2.1 Constraints and Features

In maximum entropy we use the training data to set *constraints* on the conditional distribution. Each constraint expresses a characteristic of the training data that should also be present in the learned distribution. We let any real-valued function of the document and the class be a feature, $f_i(d,c)$. Maximum entropy allows us to restrict the model distribution to have the same *expected value* for this feature as seen in the training data, $\mathcal{D}$. Thus, we stipulate that the learned conditional distribution $\mathrm{P}(c|d)$ must have the property:

$$\frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} f_i(d,c(d)) = \sum_d \mathrm{P}(d)\sum_c \mathrm{P}(c|d) f_i(d,c). \quad (1)$$

In practice, the document distribution $\mathrm{P}(d)$ is unknown, and we are not interested in modeling it. Thus, we use our training data, without class labels, as an approximation to the document distribution, and enforce the constraint:

$$\frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} f_i(d,c(d)) = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\sum_c \mathrm{P}(c|d) f_i(d,c). \quad (2)$$

Thus, when using maximum entropy, the first step is to identify a set of feature functions that will be useful for classification. Then, for each feature, measure its expected value over the training data and take this to be a constraint for the model distribution.

2.2 Parametric Form

When constraints are estimated in this fashion, it is guaranteed that a unique distribution exists that has maximum entropy. Moreover, it can be shown [Della Pietra *et al.*, 1997] that the distribution is always of the exponential form:

$$P(c|d) = \frac{1}{Z(d)}\exp(\sum_i \lambda_i f_i(d,c)), \quad (3)$$

where each $f_i(d,c)$ is a feature, λ_i is a parameter to be estimated and $Z(d)$ is simply the normalizing factor to ensure a proper probability:

$$Z(d) = \sum_c \exp(\sum_i \lambda_i f_i(d,c)). \quad (4)$$

When the constraints are estimated from labeled training data, the solution to the maximum entropy problem is also the solution to a dual maximum likelihood problem for models of the same exponential form. Additionally, it is guaranteed that the likelihood surface is convex, having a single global maximum and no local maxima. This suggests one possible approach for finding the maximum entropy solution: guess any initial exponential distribution of the correct form as a starting point; then, perform hillclimbing in likelihood space. Since there are no local maxima, this will converge to the maximum likelihood solution for exponential models, which will also be the global maximum entropy solution.

2.3 Improved Iterative Scaling

In this section, we briefly outline the derivation of improved iterative scaling (IIS), a hillclimbing algorithm for calculating the parameters of a maximum entropy classifier given a set of constraints. We also describe the algorithmic details of this procedure. A complete description and derivation of improved iterative is presented by Della Pietra *et al.* [1997]. This presentation follows that of Berger [1998].

IIS performs hillclimbing in parameter log likelihood space. Given a set of i.i.d. training data $\mathcal{D}$, we can calculate the log likelihood of an exponential model, Λ, using Equation 3:

$$\begin{aligned} l(\Lambda|\mathcal{D}) &= \log\prod_{d\in\mathcal{D}} \mathrm{P}_\Lambda(c(d)|d) \\ &= \sum_{d\in\mathcal{D}}\sum_i \lambda_i f_i(d,c(d)) - \\ &\quad \sum_{d\in\mathcal{D}} \log\sum_c \exp\sum_i \lambda_i f_i(d,c). \end{aligned} \quad (5)$$

At each step IIS must find an incrementally more likely set of parameters. Since the likelihood function is convex, if we can guarantee that IIS succeeds in improving the likelihood, then we know it will converge to the globally optimal set of parameters—those that are both the maximum likelihood solution for the parametric form, and the solution with the maximal entropy.

We start from any initial vector of parameters Λ; at each step we improve Λ, by setting it equal to $\Lambda + \Delta$, which will have a higher likelihood. Thus, at each step, we want to find a Δ such that the difference in likelihoods is positive:

$$l(\Lambda + \Delta|\mathcal{D}) - l(\Lambda|\mathcal{D}) > 0. \quad (6)$$

Using the inequality $-\log(x) \geq 1 - x$ and Jensen's inequality, we can bound this expression from below with an auxiliary function we call B:

$$\begin{aligned} l(\Lambda + \Delta|\mathcal{D}) - l(\Lambda|\mathcal{D}) \quad &\geq \quad B = \\ 1 + \sum_{d \in \mathcal{D}} \Big(&\sum_i \delta_i f_i(d, c(d)) - \\ \sum_c P_\Lambda(c|d) \exp(f^\sharp(d,c)\delta_i &\sum_i \frac{f_i(d,c)}{f^\sharp(d,c)}) \Big), \quad (7) \end{aligned}$$

where $f^\sharp(d,c)$ is the sum of all features in training instance d:

$$f^\sharp(d,c) = \sum_i f_i(d,c). \quad (8)$$

We can guarantee an increase in likelihood if we can find a Δ such that B is positive. We can find the best Δ by differentiating B with respect to the change in each parameter δ_i in turn and solving for the maxima:

$$\begin{aligned} \frac{\partial B}{\partial \delta_i} \quad = \quad &\sum_{d \in \mathcal{D}} \Big(f_i(d, c(d)) - \\ &\sum_c \mathrm{P}_\Lambda(c|d) f_i(d,c) \exp(\delta_i f^\sharp(d,c)) \Big). \quad (9) \end{aligned}$$

Note that it is straightforward to set this equal to zero and solve for the increment δ_i for each parameter λ_i. In the case where $f^\sharp(d,c)$ is constant for all d and c (as in experiments in this paper), this can be solved in closed-form. Otherwise, this can be solved with a numeric root-finding procedure, such as Newton's method. In this case, the polynomial is guaranteed to have only one positive root.

This analysis shows that we can find at each hillclimbing step changes to each λ_i that improve the model likelihood. Since the likelihood is convex, this hillclimbing is guaranteed to converge to the global maximum.

This is the foundation for the improved iterative scaling algorithm, outlined in Table 1. At each step of the iteration we need to estimate class labels $\mathrm{P}_\Lambda(c|d)$ of all documents with the current model. Then, using the class labels we calculate improved model parameters and iterate.

- **Inputs:** A collection $\mathcal{D}$ of labeled documents and a set of feature functions f_i.
- Set the constraints (Equation 2). For every feature f_i, estimate its expected value on the training documents.
- Initialize all the λ_i's to be zero.
- Iterate until convergence:
  - Calculate the expected class labels for each document with the current parameters, $P_\Lambda(c|d)$ (Equation 3).
  - For each parameter λ_i:
    - Set $\frac{\partial B}{\partial \delta_i} = 0$ and solve for δ_i (Equation 9).
    - set $\lambda_i = \lambda_i + \delta_i$
- **Output:** A text classifier that takes an unlabeled document and predicts a class label.

Table 1: An outline of the Improved Iterative Scaling algorithm for estimating the parameters for maximum entropy.

2.4 Gaussian Prior

Maximum entropy can suffer from overfitting. The constraints are estimated from labeled training data, and, like other learning algorithms, when data is sparse, overfitting can occur. With too little data, the expected value of a feature in the training data may be far from the true value. By introducing a prior on the model, overfitting can be reduced and performance improved.

To integrate a prior into maximum entropy, we use maximum a posteriori estimation for the exponential model, instead of maximum likelihood estimation. We use a Gaussian prior for the model, with the mean at zero, and a diagonal covariance matrix. This prior favors feature weightings that are closer to zero, that is, are less extreme. The prior probability of the model is just the product over the Gaussian of each feature value λ_i with variance σ_i^2:

$$\mathrm{P}(\Lambda) = \prod_i \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(\frac{-\lambda_i^2}{2\sigma_i^2}). \quad (10)$$

Integrating this prior into improved iterative scaling requires adding a single term to the derivative of B (Equation 9):

$$\begin{aligned} \frac{\partial B}{\partial \delta_i} \quad = \quad &\frac{\lambda_i + \delta_i}{-\sigma_i^2} + \sum_{d \in \mathcal{D}} \Big(f_i(d, c(d)) - \\ &\sum_c \mathrm{P}_\Lambda(c|d) f_i(d,c) \exp(\delta_i f^\sharp(d,c)) \Big). (11) \end{aligned}$$

Again, this new formula is easily solved for a maximum with a numeric root-finding procedure, like Newton's method. Chen and Rosenfeld [1999] have shown that introducing a Gaussian prior on each λ_i improves performance for language modeling tasks when sparse data causes overfitting. This paper also derives the update rule given by Equation 11.

3 Maximum Entropy for Text Classification

In order to apply maximum entropy to a domain, we need to select a set of features to use for setting the constraints. For text classification with maximum entropy, we use word counts as our features. More specifically, in this paper for each word-class combination we instantiate a feature as:

$$f_{w,c'}(d,c) = \begin{cases} 0 & \text{if } c \neq c' \\ \frac{N(d,w)}{N(d)} & \text{Otherwise,} \end{cases} \quad (12)$$

where $N(d,w)$ is the number of times word w occurs in document d, and $N(d)$ is the number of words in d.

With this representation, if a word occurs often in one class, we would expect the weight for that word-class pair to be higher than for the word paired with other classes. In most natural language tasks using maximum entropy the features are naturally binary features. In text classification, we expect that features accounting for the number of times a word occurs should improve classification. For example, naive Bayes implementations that use counts outperform implementations that do not [McCallum and Nigam, 1998]. Note that we use scaled counts as features instead of simple counts. We initially choose this representation for computational efficiency, as it lets us perform each IIS iteration in closed form. Some implications of this choice are discussed in Section 6.

One especially pleasing aspect of maximum entropy is that it does not suffer from any independence assumptions. For example, consider the phrase "Buenos Aires," where the two words almost always co-occur, and only rarely occur by themselves. Naive Bayes will double-count the evidence of this phrase. Maximum entropy, on the other hand, will discount the λ_i for each of these features such that their weight towards classification is appropriately reduced by half. This is because the constraints work over *expectations* of the counts. One implication of this freedom from independence assumptions is that bigrams and phrases can be easily added as features by maximum entropy, without worry that the features are overlapping. Experiments with such expanded features is a promising area of future work.

4 Related Work

Two other studies have been performed using maximum entropy for text classification. The first, a study by Ratnaparkhi [1998], is a very preliminary experiment. In a comparison between maximum entropy and decision trees, maximum entropy performs better at classifying the acq class in the Reuters-21578 data set. Here, binary features are used instead of counts. We generally expect that representing counts instead of binary features should enhance performance.

A recent study on feature selection and model building for maximum entropy [Mikheev, 1999] examined text classification performance on the RAPRA corpus of technical abstracts. Here, maximum entropy compares favorably to a smoothed logistic term-weighting model. Again, features are only binary valued. Interestingly, the use of pairs of words and word phrases as features improved performance.

5 Results

This section provides some preliminary empirical evidence that maximum entropy is a competitive text classification algorithm. The results are based on three different data sets.[1]

5.1 Data Sets and Protocol

The WebKB data set [Craven *et al.*, 1998] contains web pages gathered from university computer science departments. The pages are divided into seven categories: student, faculty, staff, course, project, department and other. In this paper, we use the four most populous entity-representing categories: student, faculty, course and project, all together containing 4199 pages. We did not use stemming or a stoplist. The resulting vocabulary has 23830 words.

The Industry Sector hierarchy, made available by *Market Guide Inc.* (www.marketguide.com) consists of company web pages classified in a hierarchy of industry sectors [McCallum *et al.*, 1998]. There are 6440 web pages partitioned into the 71 classes that are two levels deep in the hierarchy. In tokenizing the data we do not stem. After removing tokens that occur only once or are on a stoplist, the corpus has a vocabulary of size 29964.

The Newsgroups data set contains about 20,000 articles evenly divided among 20 UseNet discussion groups [Joachims, 1997]. Many of the categories fall into confusable clusters; for example, five 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 with no stemming. Documents containing UU-encoded segments were discarded. The resulting vocabulary, after removing words that occur only once or on a stoplist, has 57040 words.

Empirical results with maximum entropy are compared to naive Bayes [Lewis, 1998; Mitchell, 1997], a popular baseline for text classification. We use the multinomial instantiation of naive Bayes [McCallum and Nigam, 1998], which accounts for the number of times each word occurs. Two variants of multinomial naive Bayes are tested. In *scaled naive Bayes*, each word count in a document is scaled such that each document has a constant number of word occurrences. In *regular naive Bayes*,

[1] These data sets are all available on the Internet. See http://www.cs.cmu.edu/~TextLearning.

Data Set	Regular naive Bayes	Scaled naive Bayes	Basic Maximum Entropy	Maximum Entropy w/ Prior
WebKB	13.69 (2000)	13.10 (5000)	7.92 (2000)	8.08 (2000)
Industry Sector	28.97 (20000)	20.21 (29964)	21.14 (29964)	18.90 (29964)
Newsgroups	16.15 (57040)	14.43 (57040	15.77 (57040)	15.14 (57040)

Table 2: Classification error (%) of maximum entropy text classification on three data sets, compared to regular and scaled naive Bayes. Each is shown at their optimal vocabulary size, indicated in parentheses. Note that maximum entropy always outperforms regular naive Bayes, but the comparison is mixed with scaled naive Bayes.

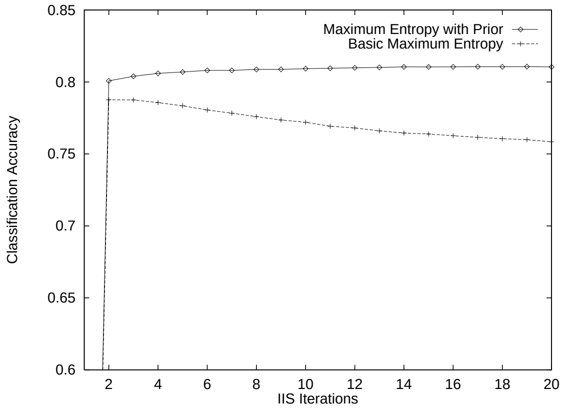

Figure 1: Accuracy over iterations of improved iterative scaling on the Industry Sector dataset with the full vocabulary, where it does best on this dataset. For basic maximum entropy, initially, accuracy is very good, and then degrades slowly, indicating the possibility of overfitting. Problems with overfitting are reduced with a Gaussian prior, and performance improves. Note the scaled vertical axis.

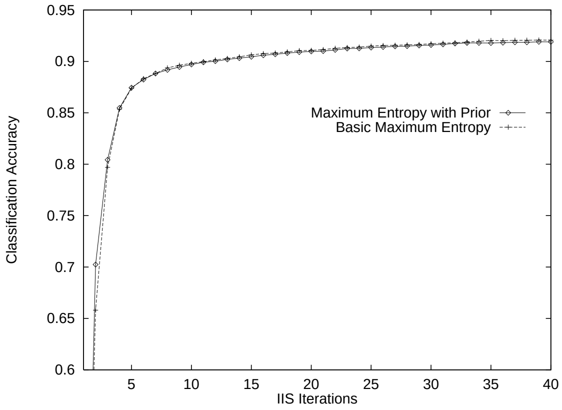

Figure 2: Accuracy over iterations of improved iterative scaling on the WebKB dataset with a 2000 word vocabulary, where it does best on this dataset. The trend here is very different than in the Industry Sector dataset. Here, accuracy continues to improve gradually over many iterations of IIS. Performance is essentially unchanged with a Gaussian prior. Note the scaled vertical axis.

word counts are left unscaled. Previous work [Nigam *et al.*, 1999] has observed that on some datasets, scaled naive Bayes outperforms regular naive Bayes.

Vocabulary selection for naive Bayes and maximum entropy is performed by taking the top words by mutual information with the class variable. This is a commonly-used technique for vocabulary selection in naive Bayes text classification [Yang and Pederson, 1997]. With maximum entropy, each feature is the normalized count of the number of times a word occurs given that the document belongs to a specific class (Equation 12). Constraints are created for all word-class pairs for which there is at least some training data. Thus, we do not constrain the expected value of a feature to be zero.

In experiments with a Gaussian prior, a single variance is chosen for all features. Choosing this variance, as well as the vocabulary size, is done by optimizing performance on the test set. In practice, these parameters can be set by cross-validation.

Maximum entropy and naive Bayes experiments are performed with ten trials of randomly selected train-test splits. For the WebKB data set, 30% of the documents are held-out for testing. For Industry Sector and Newsgroups, 35% of the documents are held-out. For Newsgroups and Industry Sector, basic maximum entropy suffers from overfitting (see results and discussion below). For this reason, 5% of the documents are used in these cases as a validation set for early stopping of IIS iterations. Other maximum entropy experiments run for a fixed number of iterations.

5.2 Experiments

Table 2 shows classification error results for each of the three algorithms on each dataset. The first two columns show performance with the two variations of naive Bayes. As an interesting aside, not that scaled naive Bayes is more accurate than regular naive Bayes on these data sets. The third column shows the performance of basic maximum entropy, without a prior. Note that in all cases, maximum entropy performs better than regular naive Bayes. In some cases, the difference is dramatic; for example on the WebKB dataset, maximum entropy provides a 40% reduction in error over naive Bayes. However, in comparison to scaled naive Bayes, the results are mixed. On WebKB, maximum entropy gives lower error, but for Industry Sector and Newsgroups, it does slightly worse.

On the two datasets where maximum entropy performs worse than scaled naive Bayes, a closer analysis of basic maximum entropy indicates that it is overfitting the training data. The bottom line in Figure 1 shows

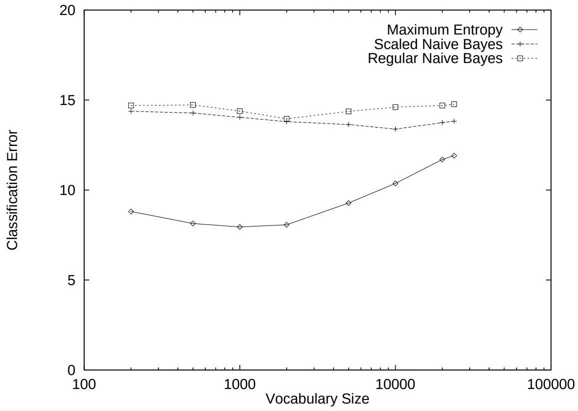

Figure 3: Classification error on the WebKB data set with different vocabulary sizes. Note the increase in error at large vocab sizes for maximum entropy, indicating the importance of feature selection.

how the accuracy of the maximum entropy classifier progresses on the Industry Sector data through the rounds of improved iterative scaling when a validation set is not used for halting. Note that best performance, 78.8%, is achieved at only the second iteration of IIS. However, by the 20th iteration of IIS, accuracy has declined to 75.8%. Similar trends appear in the 20 Newsgroups data, but not in the WebKB data (Figure 2), where maximum entropy performs better than both naive Bayes variations. These results show that basic maximum entropy is overfitting the data in cases where it is not performing well.

When maximum entropy is used with a Gaussian prior, overfitting is reduced, and performance improves. The top line in Figure 1 shows classification accuracy of maximum entropy when a prior is used. Here, performance does not degrade, and classification error is better. The fourth column of Table 2 shows classification error for these cases. In cases where overfitting was evident, error is decreased. Now, performance on the Industry Sector data set is better than scaled naive Bayes. Performance on the WebKB data set, which had no overfitting problems, is essentially unchanged.

Further analysis indicates some areas for future work. Figure 3 shows error of the classifiers across different vocabulary sizes for WebKB. Here, the error of the maximum entropy classifier increases rather suddenly after several thousand words. This shows that feature selection is an important factor for maximum entropy. In these experiments, feature selection was performed with a method that is natural for naive Bayes. In the next section, we discuss some feature selection techniques that would be more appropriate for maximum entropy.

6 Future Work

Many areas of future work remain. The results indicate that maximum entropy may be sensitive to poor feature selection. Since a feature for maximum entropy is a combination of a class and a word, there is no need to have features for all classes for a vocabulary word. For example, “professor” could be a feature for the faculty and course classes, but nor for the student class. An iterative greedy feature selection technique for maximum entropy [Della Pietra *et al.*, 1997] has been shown to create compact representations that result in good maximum entropy performance. We intend to test such an approach for text classification.

In the experiments presented here, the same Gaussian prior variance was used for all feature values. This need not be the case. For features with a large amount of training data, overfitting should not be a problem, and a large variance can be used for the prior. For features with only sparse training data, a strong prior (smaller variance) should be used. Future experiments that adjust the prior based on the amount of training data may improve our results further.

Another area of our ongoing work lies in the representation of features and constraints. In the results presented here, we use scaled counts as features. Preliminary results using unscaled counts indicate that accuracy decreases. We hypothesize that using unscaled counts hurts for long documents where repeated words are given too strong a weight. This suggests using feature functions of the form $\log(count)$ or some other sub-linear representation instead of the counts themselves.

One promising aspect of maximum entropy is that it naturally handles overlapping features. For example, we could supplement our word features with bigram, phrase, and even non-text features. Maximum entropy will not be hurt by strong independence assumptions, as would naive Bayes with these features. In future work, we will try to augment maximum entropy with expanded feature classes.

One last area of future work is a more thorough comparison of maximum entropy to other state-of-the-art text classification algorithms on several domains. Ongoing work includes direct comparisons of maximum entropy to support vector machines [Joachims, 1998], k-nearest neighbor [Yang, 1999], and RIPPER [Cohen and Singer, 1996].

In summary, maximum entropy is a technique that is popular for many other natural language tasks. Its overriding principle is one of minimal assumption (maximal entropy) that matches an intuition of how probability distributions should be estimated from data. Empirical analysis shows that maximum entropy is competitive with, and sometimes better than, naive Bayes text classification.

Acknowledgements

We thank Adam Berger for helpful and insightful discussion. This work was supported in part by the DARPA HPKB program under contract F30602-97-1-0215.

References

[Beeferman *et al.*, 1999] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1–3):177–210, 1999.

[Berger, 1998] Adam Berger. Convexity, maximum likelihood and all that. http://www.cs.cmu.edu/~aberger, 1998.

[Chen and Rosenfeld, 1999] Stanley F. Chen and Ronald Rosenfeld. A Gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, Carnegie Mellon University, 1999.

[Cohen and Singer, 1996] William W. Cohen and Yoram Singer. Context-sensitive learning methods for text categorization. In *SIGIR '96: Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–315, 1996.

[Craven *et al.*, 1998] M Craven, D DiPasquo, D Freitag, A McCallum, T Mitchell, K Nigam, and S Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the Fifteenth National Conference on Artificial Intellligence (AAAI-98)*, pages 509–516, 1998.

[Csiszár, 1996] I. Csiszár. Maxent, mathematics, and information theory. In K. Hanson and R. Silver, editors, *Maximum Entropy and Bayesian Methods*. Kluwer Academic Publishers, 1996.

[Della Pietra *et al.*, 1997] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), 1997.

[Dumais *et al.*, 1998] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of ACM-CIKM98*, 1998.

[Joachims, 1997] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML '97)*, pages 143–151, 1997.

[Joachims, 1998] Thorsten Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML-98, Tenth European Conference on Machine Learning*, pages 137–142, 1998.

[Lewis, 1998] David D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In *Machine Learning: ECML-98, Tenth European Conference on Machine Learning*, pages 4–15, 1998.

[McCallum and Nigam, 1998] Andrew McCallum and Kamal Nigam. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998. Tech. rep. WS-98-05, AAAI Press. http://www.cs.cmu.edu/~mccallum.

[McCallum *et al.*, 1998] Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Ng. Improving text clasification by shrinkage in a hierarchy of classes. In *Machine Learning: Proceedings of the Fifteenth International Conference (ICML '98)*, pages 359–367, 1998.

[Mikheev, 1999] Andrei Mikheev. Feature lattics and maximum entropy models. *Machine Learning*, 1999. To appear.

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

[Nigam *et al.*, 1999] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 1999. To appear.

[Ratnaparkhi *et al.*, 1994] Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 250–255, 1994.

[Ratnaparkhi, 1996] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Empirical Methods in Natural Language Conference*, 1996.

[Ratnaparkhi, 1998] Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, 1998.

[Rosenfeld, 1994] Ronald Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*. PhD thesis, Carnegie Mellon University, 1994.

[Sahami, 1996] Mehran Sahami. Learning limited dependence Bayesian classifiers. In *KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 335–338. AAAI Press, 1996.

[Schapire and Singer, 1999] Robert E. Schapire and Yoram Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 1999. To appear.

[Slattery and Craven, 1998] Sean Slattery and Mark Craven. Combining statistical and relational methods for learning in hypertext domains. In *Proceedings of the 8th International Conference on Inductive Logic Programming (ILP-98)*, 1998.

[Yang and Pederson, 1997] Yiming Yang and Jan O. Pederson. Feature selection in statistical learning of text categorization. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML '97)*, pages 412–420, 1997.

[Yang, 1999] Yiming Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1999. To appear.