CREATING ENSEMBLE OF DIVERSE MAXIMUM ENTROPY MODELS

Kartik Audhkhasi[1*], *Abhinav Sethy*[2], *Bhuvana Ramabhadran*[2], *Shrikanth S. Narayanan*[1]
[1]Signal Analysis and Interpretation Lab (SAIL), University of Southern California, Los Angeles, CA
[2]IBM T J Watson Research Center, Yorktown Heights, NY
audhkhas@usc.edu, asethy@us.ibm.com, bhuvana@us.ibm.com, shri@sipi.usc.edu

ABSTRACT

Diversity of a classifier ensemble has been shown to benefit overall classification performance. But most conventional methods of training ensembles offer no control on the extent of diversity and are meta-learners. We present a method for creating an ensemble of diverse maximum entropy (∂MaxEnt) models, which are popular in speech and language processing. We modify the objective function for conventional training of a MaxEnt model such that its output posterior distribution is diverse with respect to a reference model. Two diversity scores are explored – KL divergence and posterior cross-correlation. Experiments on the CoNLL-2003 Named Entity Recognition task and the IEMOCAP emotion recognition database show the benefits of a ∂MaxEnt ensemble.

Index Terms— Maximum entropy model, classifier diversity

1. INTRODUCTION

Ensembles of multiple experts have out-performed single experts in many pattern classification tasks. Well-known examples include the Netflix Challenge [1], the 2009 KDD Orange Cup [2] and the DARPA GALE program [3]. Dietterich [4] notes three reasons which can explain this. First, an ensemble can potentially have lower generalization error as compared to individual classifiers. Second, the training of most state of the art classifiers (e.g. neural networks) involves solving a non-convex optimization problem. Thus, while the individual classifiers can get stuck in local optima, the ensemble has a better chance to come close to the global optima. Finally, the true decision boundary for the problem at hand may be too complex for a single classifier and an ensemble may better approximate it.

Two popular methods for training classifier ensembles are bagging (bootstrap aggregating) [5] and AdaBoost (adaptive boosting) [6]. Consider a training set $\mathcal{T}$ containing N pairs of feature vectors and target variables, $\{(x_n, y_n)\}_{n=1}^{N}$. Bagging proceeds by sampling $\mathcal{T}$ with replacement and creating M bootstrapped data sets $\mathcal{T}_1, ..., \mathcal{T}_M$. The m^{th} classifier (or regressor) is then trained on $\mathcal{T}_m$. Given a test feature vector x, results from the M experts are averaged to yield the estimated target variable. Breiman uses a bias-variance decomposition to prove that in case of regression, the mean squared error of the average regressor is less than or equal to the average mean squared error over the individual regressors. The second method, AdaBoost, works by sequentially training the classifiers in the ensemble. The training data for the m^{th} classifier, $\mathcal{T}_m$, is created by weighted sampling from $\mathcal{T}$, where greater probability mass is assigned to the instances which are misclassified by classifiers $1, ..., m-1$. Freund and Schapire have derived an upper bound on the training error of the ensemble, which indicates that increasing the size of the ensemble in AdaBoost reduces the training error towards zero. AdaBoost can also be viewed as minimizing the exponential loss between the training and predicted label.

As noted in [7], the diversity of classifiers in an ensemble is crucial for its overall performance. Ueda and Nakano [8] consider diversity in an ensemble of regressors and derive a bias-variance-covariance decomposition for the average regressor's mean squared error. The mean squared error reduces as the pairwise diversity between individual regressors (accounted for by the covariance term) increases. Tumer and Ghosh [9] extend the analysis to classification by treating it as regression over the class posteriors. The additional error of the ensemble over the Bayes optimal error is shown to be dependent on the correlation coefficient between class posteriors.

A typical approach to introduce diversity is to use radically different classifiers and/or feature sets. However, this does not offer explicit control on the extent of diversity achieved. Bagging and boosting also suffer from this issue, and require weak/unstable base classifiers for giving a substantial performance gain. Inspite of the evidence linking diversity and ensemble performance, only a few works deal with explicity creating diverse classifier ensembles. Negative Correlation Learning [10] involves decorrelating errors from the individual neural networks as part of their training. Another work is DECORATE [11], a meta-learner where the ensemble is built incrementally with each successive classifier trained on a mix of artificial and natural data. Artificial training instances are labeled contrary to the opinion of the current ensemble.

This paper focusses on training diverse maximum entropy (MaxEnt) models. MaxEnt models are state of the art classifiers in many domains, especially speech and language processing. They possess several desirable properties such as flexibility in adding new features, scalable training, easy parameter estimation and minimal assumptions about the posteriors. The next section discusses our approach for training a diverse MaxEnt (∂MaxEnt) ensemble. We present experiments and analysis on the CoNLL-2003 Named Entity Recognition task and the IEMOCAP emotion recognition database in section 3. Conclusions and scope for future work are presented in section 4.

2. TRAINING A ∂MAXENT ENSEMBLE

We first review the standard MaxEnt model to set up the notation. Let $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ denote the feature vector and class label respectively. The maximum entropy principle aims to find a probability distribution $P(y|x)$ with maximum entropy subject to the following first order moment constraints for training data $\mathcal{T}$:

$$\sum_{n=1}^{N} E_P\{f_i(x_n, y)\} = \sum_{n=1}^{N} f_i(x_n, y_n) \quad \forall i \in \{1, ..., F\} \qquad (1)$$

where f_i is the i^{th} feature - an arbitrary function of x and y. E_P denotes the expectation with respect to $P(y|x_n)$. This problem can

[*] This work was done during a summer internship at IBM T J Watson Research Center, Yorktown Heights, NY.

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

be solved by Lagrange's method, and the resulting distribution is:

$$P_\Lambda(y|x) = \frac{\exp(\sum_{i=1}^{F} \lambda_i f_i(x,y))}{\sum_{y\in\mathcal{Y}} \exp(\sum_{i=1}^{F} \lambda_i f_i(x,y))} \tag{2}$$

where λ_i are the Lagrange multipliers. The log-likelihood function of the MaxEnt model over training data $\mathcal{T}$ is[1]:

$$\mathcal{L}(\Lambda) = \sum_{n=1}^{N}(\sum_{i=1}^{F} \lambda_i f_i(x_n, y_n) - \log Z(x_n)) \tag{3}$$

where $Z(x_n)$ is the normalization sum in the denominator of Eq. 2. We note that $\mathcal{L}(\Lambda)$ is concave in $\lambda_i \forall i$. Hence a simple gradient ascent, Newton-Raphson or a quasi-Newton method (such as L-BFGS [12]) can be used to find the maximum likelihood parameter estimates. The gradient of $\mathcal{L}(\Lambda)$ is given as:

$$\frac{\partial \mathcal{L}(\Lambda)}{\partial \lambda_i} = \sum_{n=1}^{N} \Big(f_i(x_n, y_n) - E_P\{f_i(x_n, y)\}\Big) \tag{4}$$

Our task is to train an ensemble of diverse MaxEnt models. We first study the simpler case of training a MaxEnt model which fits the data well but is diverse with respect to a reference model $Q_{\Lambda'}$. A natural way to achieve this is to introduce a diversity term in the log-likelihood function as follows:

$$\mathcal{L}_{tot}(\Lambda) = \mathcal{L}(\Lambda) + \alpha\mathcal{D}(P_\Lambda, Q_{\Lambda'}) \tag{5}$$

where $\alpha \geq 0$ is the diversity weight and $\mathcal{D}(P_\Lambda, Q_{\Lambda'})$ is the diversity between the two models. As is noted in [13], there are multiple ways to capture diversity between two classifiers. We use two intuitive diversity scores - the Kullback-Leibler (KL) divergence between posterior distributions and negative posterior cross-correlation.

2.1. KL Divergence Diversity

The KL divergence from $Q_{\Lambda'}(y|x_n)$ to $P_\Lambda(y|x_n)$ is the following ensemble average:

$$KL_n(Q_{\Lambda'}||P_\Lambda) = \sum_{y\in\mathcal{Y}} Q_{\Lambda'}(y|x_n) \log \frac{Q_{\Lambda'}(y|x_n)}{P_\Lambda(y|x_n)} \tag{6}$$

We did not use $KL_n(P_\Lambda||Q_{\Lambda'})$ due to difficulty in interpreting its gradient. Adding this expectation over all instances in the training data, the modified log-likelihood becomes:

$$\mathcal{L}_{tot}(\Lambda) = \mathcal{L}(\Lambda) + \alpha \sum_{n=1}^{N} KL_n(Q_{\Lambda'}||P_\Lambda) \tag{7}$$

While $\mathcal{L}(\Lambda)$ is concave in Λ, $KL_n(Q_{\Lambda'}||P_\Lambda)$ is convex, attaining a minimum value of 0 at $\Lambda = \Lambda'$. Thus the overall objective function is neither concave nor convex and one can only hope to obtain locally optimal estimates of Λ. Furthermore, KL divergence can potentially approach $+\infty$, making the objective function unbounded. The gradient of $\mathcal{L}_{tot}(\Lambda)$ can be written as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{tot}(\Lambda)}{\partial \lambda_i} = \sum_{n=1}^{N} \Big(& f_i(x_n, y_n) \\ & - [(1-\alpha)E_P\{f_i(x_n, y)\} + \alpha E_Q\{f_i(x_n, y)\}]\Big)\end{aligned} \tag{8}$$

This expression is the same as for a conventional MaxEnt model (Eq. 4), except that a linear combination of the feature expectation under P_Λ and $Q_{\Lambda'}$ is taken. Increasing α has the effect of increasing the weight on the expectation from the reference model ($Q_{\Lambda'}$). While it seems that KL divergence should succeed in achieving diversity between the models, it can be easily shown that this may not be the case in practice. Let the reference model $Q_{\Lambda'}$ be trained on data set $\mathcal{T}$ using features $\{f_i\}_{i=1}^{F}$ by maximizing $\mathcal{L}(\Lambda')$. Upon convergence of its training, the gradient of $\mathcal{L}(\Lambda')$ will be zero. Hence:

$$\sum_{n=1}^{N} E_Q\{f_i(x_n, y)\} = \sum_{n=1}^{N} f_i(x_n, y_n) \quad \forall i \in \{1, ..., F\} \tag{9}$$

If P_Λ is trained to be diverse with respect to $Q_{\Lambda'}$ by maximizing $\mathcal{L}_{tot}(\Lambda)$ using the same data and feature set, we can substitute the above equation in Eq. 8 and arrive at the following result:

$$\frac{\partial \mathcal{L}_{tot}(\Lambda)}{\partial \lambda_i} = (1-\alpha)\sum_{n=1}^{N} \Big(f_i(x_n, y_n) - E_P\{f_i(x_n, y)\}\Big) \tag{10}$$

Hence the gradients for a MaxEnt and ∂MaxEnt are the same upto a scalar multiple. At a local optima, the parameter estimates will satisfy the same constraint as in the case of a conventional MaxEnt model. This problem with KL divergence can be mitigated to some extent by using distinct training sets or features for $Q_{\prime}$ and P_Λ. However it necessitates the search for another diversity score. The next subsection introduces posterior cross-correlation to this end.

2.2. Posterior Cross-Correlation (PCC) Diversity

Making a simplistic assumption, consider independent random variables $y_P \sim P_\Lambda(y|x)$ and $y_Q \sim Q_{\Lambda'}(y|x)$. The conditional probability of them being unequal is:

$$Pr\{y_P \neq y_Q|x\} = 1 - \sum_{y\in\mathcal{Y}} P_\Lambda(y|x)Q_{\Lambda'}(y|x) \tag{11}$$

Thus, negative cross-correlation between the two posterior distributions is a natural diversity score. The modified log-likelihood function can be written as follows:

$$\mathcal{L}_{tot}(\Lambda) = \mathcal{L}(\Lambda) - \alpha \sum_{n=1}^{N}\sum_{y\in\mathcal{Y}} P_\Lambda(y|x_n)Q_{\Lambda'}(y|x_n) \tag{12}$$

This objective function is again neither convex nor concave. However, unlike KL divergence, it has the following finite bounds:

$$\min_{y\in\mathcal{Y}} Q_{\Lambda'}(y|x_n) \leq \sum_{y\in\mathcal{Y}} P_\Lambda(y|x_n)Q_{\Lambda'}(y|x_n) \leq \max_{y\in\mathcal{Y}} Q_{\Lambda'}(y|x_n)$$

The gradient can be shown to be equal to:

$$\begin{aligned}\frac{\partial \mathcal{L}_{tot}(\Lambda)}{\partial \lambda_i} = \sum_{n=1}^{N} f_i(x_n, y_n) - \sum_{n=1}^{N}[&(1 - Z_{PQ}(x_n)\alpha)E_P\{f_i(x_n, y)\} \\ &+ Z_{PQ}(x_n)\alpha E_{PQ}\{f_i(x_n, y)\}]\end{aligned} \tag{13}$$

where $PQ_{\Lambda,\Lambda'}(y|x_n)$ is the normalized product distribution:

$$PQ_{\Lambda,\Lambda'}(y|x_n) = \frac{Q_{\Lambda'}(y|x_n)P_\Lambda(y|x_n)}{Z_{PQ}(x_n)} \tag{14}$$

and $Z_{PQ}(x_n) = \sum_{y\in Y} Q_{\Lambda'}(y|x_n)P_\Lambda(y|x_n)$ is the normalization constant. The above gradient is similar to the one for KL divergence

[1] Penalizing this function by the L_1 and L_2 norms of Λ has been empirically shown to give performance benefits.

except for two modifications – expectation with respect to the product distribution is used instead of $Q_{\Lambda'}$ and the linear combination weights become dependent on the instance x_n. Thus, for instances where $Z_{PQ}(x_n)$ is high (i.e. the current and reference model posteriors are highly correlated), more weight is given to the expectation with respect to the product distribution. In effect, the model deviates more from the ML estimate in these instances. Also, in case of identical training sets and features for the two models, the gradient does not reduce to the standard MaxEnt model's gradient. Till now, we have discussed a method to train a MaxEnt model P_Λ to be diverse with respect to another MaxEnt model $Q_{\Lambda'}$. The next subsection discusses one possible method in which an ensemble of $M \geq 2$ ∂MaxEnt models can be trained.

2.3. Sequential Training of a ∂MaxEnt Ensemble

Consider the training of an ensemble of M MaxEnt classifiers $P_{\Lambda_1}, ..., P_{\Lambda_M}$ with corresponding training sets $\mathcal{T}_1, ..., \mathcal{T}_M$. A simple strategy is to train the ensemble sequentially. Let MaxEnt($\mathcal{T}$) denote a function which trains a conventional MaxEnt model on $\mathcal{T}$ and returns the parameters Λ. Let ∂MaxEnt($\mathcal{T}$,$Q_{\Lambda'}$,α,Λ^0) denote a function which trains a ∂MaxEnt model on $\mathcal{T}$ with respect to $Q_{\Lambda'}$ using α as the diversity weight and Λ^0 as the initial value of the parameters. The sequential training process is as follows:

- Train model 1: $\Lambda_1 = $ MaxEnt($\mathcal{T}_1$).
- For $m = 2 \rightarrow M$
  - Initialize: $\Lambda_m^0 = $ MaxEnt($\mathcal{T}_m$).
  - Interpolate models $1, ..., m-1$:
    $Q(y|x_n) = \frac{1}{m-1} \sum_{j=1}^{m-1} P_{\Lambda_j}(y|x_n)$
    $\forall y \in \mathcal{Y}, n \in \{1, ..., |\mathcal{T}_m|\}$.
  - Train model m: $\Lambda_m = $ ∂MaxEnt($\mathcal{T}_m$,Q,α,Λ_m^0)

Since the objective function is no longer concave, we train a ∂MaxEnt model in two passes. The first pass finds the ML estimates of the parameters. The second pass performs ∂MaxEnt training using the ML parameters as the starting point. This ensures that L-BFGS converges at a local maxima which is not too far from the ML estimate while ensuring diversity. α is tuned based on F1 score on a development set. During the test phase, labels from all classifiers in the ensemble are fused by simple plurality. More sophisticated ways of classifier fusion were not experimented with since they are not the focus of this paper.

3. EXPERIMENTS AND RESULTS

The CoNLL-2003 Named Entity Recognition (NER) Task has four types of named entities - persons, locations, organizations and miscellaneous [14]. The English task consists of news wire stories from the Reuters corpus between August 1996 and August 1997. We used binary features from Stanford's NER system which include word identity, POS tags, word character N-grams etc [15]. Original training, development and evaluation sets were used. Performance was measured in terms of the F1 score for named entity detection [14].

Identical training sets	Dev set	Eval set
1 MaxEnt model	91.22	86.75
5 KL-∂MaxEnt ($\alpha_d = 1.66$)	91.31	86.73
($\alpha_e = 1.58$)	-	86.77
5 PCC-∂MaxEnt ($\alpha_d = 1.46$)	**91.70**	87.05
($\alpha_e = 1.27$)	-	**87.25**
Bagged training sets		
5 MaxEnt models	90.49	85.98
5 KL-∂MaxEnt ($\alpha_d = 0.24$)	90.62	86.15
($\alpha_e = 0.13$)	-	86.34
5 PCC-∂MaxEnt ($\alpha_d = 1.45$)	**91.21**	**86.74**
($\alpha_e = 1.45$)	-	**86.74**

Table 1. NER F1 scores for 5 MaxEnt and ∂MaxEnt models using KL/PCC diversity. α_d and α_e denote the best values of α tuned on the development and evaluation set respectively. Values in bold indicate a statistically significant improvement over the MaxEnt ensemble at the 5% level using McNemar's test.

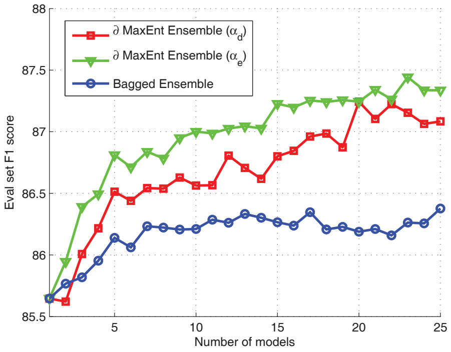

Fig. 1. F1 score on the NER evaluation set for PCC-∂MaxEnt and bagged ensembles of increasing size. α_d was tuned on the development set and α_e on the evaluation set.

Table 1 shows the F1 scores for ensembles of 5 conventional MaxEnt and ∂MaxEnt models using the two diversity scores. Two cases are considered – when the 5 training sets are identical and when they are created by bagging. We can observe that for identical training sets, KL divergence gives almost the same performance as 1 MaxEnt model. The minute difference is due to deliberate smoothing of the posterior distributions to prevent KL divergence from becoming indeterminate. On the other hand, PCC-based ∂MaxEnt models give an appreciable increase in performance. In the case of bagged training sets, KL divergence is able to achieve a statistically insignificant performance gain over 5 MaxEnt models. However, PCC-based ∂MaxEnt models still perform significantly better. We note that in [16], gradient boosting with 10000 2-level decision trees and Newton-Raphson optimization of the exponential loss was shown to give a similar gain over a MaxEnt model.

Since PCC performs significantly better than KL divergence, we analyse it further. Figure 1 shows the F1 score on the evaluation set with an increasing number of models (1 to 25). The performance of bagging saturates much earlier than the ∂MaxEnt ensemble. Thus the relative performance improvement of the ∂MaxEnt ensemble increases as the number of models is increased. The performance for α_e indicates an upper bound on the performance for the ∂MaxEnt ensemble. As a final analysis of the ∂MaxEnt model with PCC diversity, Figure 2 shows the variation of the development set F1 score and average log-likelihood for an ensemble of 5 models with increasing α. The F1 score increases with α until around $\alpha = 1.45$, after which it starts decreasing again. Furthermore, its behaviour becomes more variable with increasing α because the optimization problem is become more non-concave. It is interesting to note that the log-likelihood remains practically constant until $\alpha = 1$, while the F1 score increases significantly over the same range. The drop in log-likelihood from $\alpha = 1$ to 1.45 does not adversely impact the

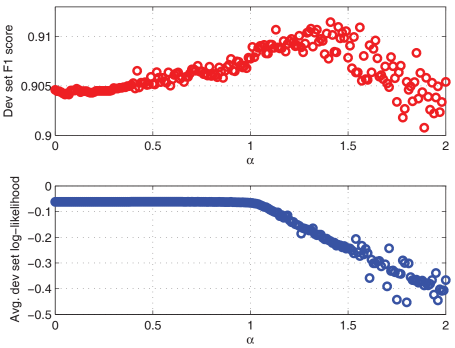

Fig. 2. F1 score and average log-likelihood for the NER development set with increasing diversity weight α. Five ∂MaxEnt models were used with PCC diversity on bagged data.

Identical training sets	Dev set	Eval set
1 MaxEnt model	43.79	44.64
5 PCC-∂MaxEnt ($\alpha_d = 0.33$)	**48.73**	**48.10**
($\alpha_e = 0.33$)	-	**48.10**
Bagged training sets		
5 MaxEnt models	43.01	43.65
5 PCC-∂MaxEnt ($\alpha_d = 0.52$)	**46.74**	**46.09**
($\alpha_e = 0.68$)	-	**47.09**

Table 2. Weighted F1 scores for emotion classification with 5 models on the IEMOCAP database.

performance.

Next, we conducted emotion classification experiments on the IEMOCAP database [17]. It is an acted, multimodal and multi-speaker database consisting of dyadic sessions where actors are asked to elicit emotional expressions. Each session was labeled by multiple human evaluators in terms of 4 categorical emotions - {angry, happy, sad, neutral}. The multiple labels were fused using simple plurality and sessions where a tie occured were excluded. A total of 5498 sessions were used, and 385 acoustic-prosodic features from the OpenSMILE toolkit [18] were extracted. These included pitch, energy, Mel-filter bank coefficients and their per-session statistics. Table 2 shows the classification performance. The ∂MaxEnt model ensemble performs significantly better than 1 MaxEnt model trained on the entire data and 5 MaxEnt models trained on bagged data. With 25 models PCC-∂MaxEnt models, we get an additional improvement of approximately 1-2%. This shows the benefit of using the ∂MaxEnt ensemble on a more difficult classification task with continuous features.

4. CONCLUSION AND SCOPE FOR FUTURE WORK

This paper presented a method to create diverse ensembles of MaxEnt models. Two intuitive diversity scores were explored - KL divergence and negative posterior cross-correlation. Experiments conducted on two classification tasks (the CoNLL-2003 Named Entity Recognition Task and the IEMOCAP emotion classification database) show the advantages of training a ∂MaxEnt ensemble. It was demonstrated that under reasonable assumptions, KL divergence achieves no gain in performance, while posterior cross-correlation performs significantly better. There are multiple directions for future work. Introduction of a diversity term in the standard MaxEnt model objective function made it non-concave – an undesirable property for optimization. We need to explore ways to train diverse models while retaining concavity. Second, since gradient boosting shows a similar gain over a MaxEnt model (albeit with thousands of models in the ensemble), the link between popular variants of boosting and ensemble diversity needs to be explored. Finally, insight into the choice of diversity scores for a given ensemble and database is required.

5. REFERENCES

[1] J. Bennett and S. Lanning, “The Netflix Prize,” in *Proceedings of KDD Cup and Workshop*, 2007, pp. 35–38.

[2] A. Niculescu-Mizil et al., “Winning the KDD Cup Orange Challenge with ensemble selection,” in *KDD Cup and Workshop in conjunction with KDD*, 2009.

[3] G. Saon et al., “The IBM 2008 GALE Arabic speech transcription system,” in *Proc. ICASSP*, 2010, pp. 4378–4381.

[4] T. Dietterich, “Ensemble methods in machine learning,” *Multiple classifier systems*, pp. 1–15, 2000.

[5] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[6] Y. Freund and R. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *Computational learning theory*. Springer, 1995, pp. 23–37.

[7] L.I. Kuncheva, *Combining pattern classifiers: methods and algorithms*, Wiley-Interscience, 2004.

[8] N. Ueda and R. Nakano, “Generalization error of ensemble estimators,” in *Proc. ICNN*, 1996, vol. 1, pp. 90–95.

[9] K. Tumer and J. Ghosh, “Analysis of decision boundaries in linearly combined neural classifiers,” *Pattern Recognition*, vol. 29, no. 2, pp. 341–348, 1996.

[10] Y. Liu and X. Yao, “Ensemble learning via negative correlation,” *Neural Networks*, , no. 10, pp. 1399–1404, 1999.

[11] P. Melville and R.J. Mooney, “Constructing diverse classifier ensembles using artificial training examples,” in *Proc. IJCAI*, 2003, pp. 505–510.

[12] D.C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[13] L.I. Kuncheva and C.J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[14] E.F. Tjong Kim Sang and F. De Meulder, “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition,” in *Proc. CoNLL*, 2003, pp. 142–147.

[15] J.R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by Gibbs sampling,” in *Proc. ACL*, 2005, pp. 363–370.

[16] B. Zhang, A. Sethy, T. N. Sainath, and B. Ramabhadran, “Application specific loss minimization using gradient boosting,” in *Proc. ICASSP*, 2011, pp. 4880–4883.

[17] C. Busso et al., “IEMOCAP: Interactive Emotional Dyadic Motion Capture Database,” *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[18] F. Eyben, M. Wollmer, and B. Schuller, “OpenSMILE: the munich versatile and fast open-source audio feature extractor,” in *Proc. Intl. Conf. on Multimedia*, 2010, pp. 1459–1462.