Investigations on Features for Log-Linear Acoustic Models in Continuous Speech Recognition

S. Wiesler, M. Nußbaum-Thom, G. Heigold, R. Schlüter, H. Ney

RWTH Aachen University
Chair of Computer Science 6 – Computer Science Department
D-52056 Aachen, Germany
{wiesler, nussbaum, heigold, schluter, ney}@informatik.rwth-aachen.de

***Abstract*—Hidden Markov Models with Gaussian Mixture Models as emission probabilities (GHMMs) are the underlying structure of all state-of-the-art speech recognition systems. Using Gaussian mixture distributions follows the generative approach where the class-conditional probability is modeled, although for classification only the posterior probability is needed. Though being very successful in related tasks like Natural Language Processing (NLP), in speech recognition direct modeling of posterior probabilities with log-linear models has rarely been used and has not been applied successfully to continuous speech recognition. In this paper we report competitive results for a speech recognizer with a log-linear acoustic model on the Wall Street Journal corpus, a Large Vocabulary Continuous Speech Recognition (LVCSR) task. We trained this model from scratch, i.e. without relying on an existing GHMM system. Previously the use of data dependent sparse features for log-linear models has been proposed. We compare them with polynomial features and show that the combination of polynomial and data dependent sparse features leads to better results.**

I. Introduction

Statistical speech recognition systems search for the most likely word sequence given an acoustic observation vector. In all state-of-the-art speech recognition systems Hidden Markov Models (HMMs) are used for the modeling of these probabilities. The concept of hidden state sequences makes HMMs invariant with respect to local time distortions. Every state sequence corresponds to a word sequence. In a (Viterbi-) recognition the most likely state sequence

$$\hat{s}_1^T = \underset{s_1^T}{\operatorname{argmax}}\, p(s_1^T, x_1^T) \tag{1}$$

is determined. Here $x_1^T = (x_1, \ldots, x_T)$ denotes the acoustic vector sequence and $s_1^T = (s_1, \ldots, s_T)$ a state sequence. Because of conditional independence assumptions of HMMs the joint probability $p(s_1^T, x_1^T)$ breaks down in a product of emission probabilities $p(x_t|s_t)$ and transition probabilities $p(s_t|s_{t-1})$:

$$p(s_1^T, x_1^T) = \prod_{t=1}^{T} p(x_t|s_t)p(s_t|s_{t-1}). \tag{2}$$

This identity enables the efficient solution of the search problem (1) with dynamic programming.
In training the parameters of these distributions are determined. In speech recognition the transition probabilities consist of a factor, which penalizes time distortions, and a language model factor which is trained separately from the acoustic model. Usually in speech recognition the time distortion penalty is set manually and not trained.
In state-of-the-art systems flexible distributions such as Gaussian Mixtures are used for modeling the emission probabilities. GHMMs can be trained efficiently according to the Maximum Likelihood (ML) criterion with the Expectation-Maximization algorithm (EM) [1]. In order to improve discrimination between states discriminative training has become a standard technique of all state-of-the-art speech recognition systems. Widely used training criteria are the Maximum Mutual Information Criterion (MMI) [2] and the Minimum Phone Error Criterion (MPE) [3].
The success of discriminative training suggests the direct modeling of posterior probabilities, because it is inherently discriminative. Neural networks and Support Vector Machines (SVMs) are popular direct models for many pattern recognition tasks. In order to incorporate them into HMM speech recognizers, the *hybrid approach* has been proposed [4], [5], where the direct model is used to estimate the posterior probability $p(s_t|x_t)$. The prior probability $p(s_t)$ can easily be calculated as the relative frequency in an existing alignment and $p(x_t)$ is a constant with respect to the maximization problem (1). Hence, according to Bayes rule

$$p(x_t|s_t) = p(s_t|x_t)\frac{p(x_t)}{p(s_t)}, \tag{3}$$

the emission probabilities in (2) can be replaced by $p(s_t|x_t)/p(s_t)$. From a statistical point of view posterior probabilities are much easier to estimate than emission probabilities, since the distribution of an observation given the state may have a complex structure even if the decision boundaries between two states behave very regularly. In contrast to other direct modeling approaches, in the hybrid approach the structure of an existing HMM speech recognizer can be fully retained.
In many NLP tasks log-linear modeling of posterior probabilities has been very successful. In this paper we study how to apply these methods to continuous speech recognition. Up to now, other authors have applied log-linear models successfully only to phone recognition tasks [6], [7] or to

digit recognition [8]. Kuo and Gao [7] considered a word recognition task, too, but they used a Maximum Entropy Markov Model [9], which does not follow the hybrid approach and is not suited to incorporate language model information. Besides, for efficiency reasons they use rank based features whereas we directly use MFCC features and probabilities. In previous studies an existing GHMM has always been used as a starting point for the log-linear training. It can either serve as an initialization for the optimization [8] or provides the time alignment used in training [6], [7]. We avoid this and train a log-linear model from scratch. In [6], the use of data dependent sparse features for the log-linear model is proposed. In this paper we show that combining them with polynomial features improves the recognition rate significantly.

We chose the Wall Street Journal corpus (WSJ0) to evaluate our proposed method. This task belongs to LVCSR and therefore requires the use of (context dependent) phoneme models. On the other hand its size is moderate, so performing a large number of comparative experiments is feasible.

The structure of the remaining paper is as follows. In Section II we present the log-linear model and describe the different types of features we investigated. Section III describes our experimental results in detail. Finally, in Section IV we discuss our results.

II. The log-linear model

Let $X \subset \mathbb{R}^D$ be the feature space and $\mathcal{S}$ a finite set of classes. A *log-linear model* with parameters $\Lambda = (\lambda_{s,i})_{s,i} \in \mathbb{R}^{|\mathcal{S}|\times n}$ is a model for posterior probabilities of the form

$$p_\Lambda(s|x) = \frac{\exp\Big(\sum_{i=1}^{n} \lambda_{s,i} f_i(x)\Big)}{\sum_{\tilde{s}} \exp\Big(\sum_{i=1}^{n} \lambda_{\tilde{s},i} f_i(x)\Big)}, \tag{4}$$

where the components of

$$f : X \to \mathbb{R}^n,\ x \mapsto (f_1(x), \ldots, f_n(x))$$

are called *feature functions*. Note that usually log-linear models are defined with feature functions depending on the class but class-independent parameters, but this definition is equivalent to the one given here.

There are two possibilities for the application of log-linear models to speech recognition. One way is to define a log-linear model on *sentence level*, i.e. for the whole acoustic sequence x_1^T [6]. These models are called Conditional Random Fields (CRF) [10]. The other possibility is to define the log-linear model on *frame level*. In Equation (4) this means x corresponds to an acoustic vector at a fixed time and s to a state of the HMM. In this paper we follow the latter approach. Some studies indicate that the CRF-approach yields slightly better results (see e.g. [8] for a comparison). In the CRF-approach the summation in (4) goes over all state sequences. In LVCSR word lattices have to be used to approximate the sum. For a training from scratch this is not possible, because word lattices are not available. In framewise modeling an existing time alignment is needed, which can either be calculated by means of an existing GHMM model or training starts with a linear time alignment. Notice that modeling on sentence level does not allow the use of better features than on frame level, because for efficiency the features of sentencewise models have to be restricted to local dependencies. Framewise training is more efficient and conceptually simpler than sentencewise training. Furthermore, the convex frame based training criterion shows better convergence behavior than the conventional lattice based training criterion.

The choice of the training criterion is an important issue. In principle all training criteria depending on a posterior probability can be used, but often the *Maximum Mutual Information criterion* (MMI) is regarded as a natural training criterion for log-linear models. An important property of log-linear models is that their optimization with respect to the MMI criterion is a convex problem. Moreover, adding an ℓ_2-regularization term to the objective function leads to a strictly convex problem. This implies there is only a single global optimum and therefore the estimated model parameters neither depend on the optimization method nor on the starting point of iterative optimization algorithms. In addition, the regularization improves the generalization abilities of the model. A further refinement of the MMI criterion is to assign a class specific weight to each observation. Given a sequence of training samples $(s_t, x_t)_{t=1,..,T}$, the resulting training criterion maximizes

$$F : \mathbb{R}^{|\mathcal{S}|\times n} \to \mathbb{R},\ \Lambda \mapsto \sum_{t=1}^{T} w_{s_t} \ln p_\Lambda(s_t|x_t) - \frac{C}{2}\|\Lambda\|_2^2, \tag{5}$$

where $C > 0$ is the regularization constant and $(w_1, \ldots, w_{|\mathcal{S}|}) \in \mathbb{R}_+^{|\mathcal{S}|}$ are the weights.

For the application of log-linear modeling suitable feature functions have to be chosen, which is the main topic of this paper. Bayes rule for log-linear models yields linear decision boundaries in $f(X)$ but not in the feature space X. By selecting suitable feature functions nonlinear decision boundaries in X can be modelled. This is similar to the utilization of kernels in SVMs.

From another point of view, the estimation of posterior probabilities with a log-linear model, is an approximation problem. The ideal feature would be the logarithm of the true posterior probability. Hence, we have to find features, such that with high probability with respect to x, the linear span of the feature functions is close to the logarithm of the true posterior probability.

A. Polynomial Features

The first type of features we consider are polynomial features. A polynomial feature function of degree k is a monomial of order k, i.e. ,

$$f_{(d_1,\ldots,d_k)} : X \to \mathbb{R},\ x \mapsto x_{d_1} \cdot \ldots \cdot x_{d_k},$$

with $(d_1, \ldots d_k) \in \{1, \ldots, D\}^k$. Using polynomial feature functions in a log-linear model corresponds to the use of polynomial kernels in SVMs.

A log-linear model with polynomial features up to order one is denoted as a *first order model*, a model with features up to order two as a *second order model* and so on. Polynomial models can be interpreted easily from a statistical point of view. A first order model is the posterior form of a generative Gaussian model with tied covariance matrices, a second order model corresponds to a Gaussian model with class specific covariance matrices. A third order model also takes class specific skewnesses of the distributions into account [11].
In principle Weierstrass' approximation theorem [12] states that any continuous function can be approximated arbitrarily well by polynomials on a compact set, which could indicate that polynomial feature functions are a good choice. In practice this result is not helpful because it does not provide information about the order of the polynomial which is necessary to achieve a certain accuracy. But the use of high order polynomial feature functions is prohibitive, since the number of polynomial feature functions grows exponentially in the order of the model. This results in a time and memory consuming training procedure because of the number of components that have to be optimized. First and second order models can be trained efficiently, training third order models is still reasonable, but going beyond third order is yet not feasible.
Polynomials are global feature functions, that means every function value depends on all data points. This makes polynomial features inflexible and slow to train as well. These problems are addressed with the use of *data dependent sparse features*.

B. Data dependent sparse features

The densities of Gaussian Mixture Models (GMMs) trained with the EM-algorithm correspond to localized dense subsets of the feature space. This makes GMMs data dependent and therefore provides good approximation abilities. To carry over this idea to the log-linear framework we can apply the EM-algorithm to estimate a single Gaussian mixture distribution for the marginal probability $p(x)$

$$p(x) = \sum_{l=1}^{L} p(l)p(x|l),$$

where L is the number of densities. In the log-linear model we use every density $p(x|l)$ of the GMM to define a feature function

$$f_l(x) = p(l|x) = \frac{p(l)p(x|l)}{\sum_{l'} p(l')p(x|l')},\ l = 1, \ldots, L.$$

The normalization of the feature functions is especially important in combination with other features, because it prevents one type of features to dominate the other. Because of the exponential decay of Gaussian densities, most feature functions will be very close to zero for fixed x. Nearly no information is lost, when the values of the feature functions are represented by n-best shortlists. These features have been proposed by Hifny in [6]. We follow this approach and modify it slightly. Instead of using an n-best shortlist we introduce a small threshold, such that the feature function is set to zero if $p(l|x)$ goes below the threshold. In the following we refer to this type of features as *clustering features*. These features correspond to radial basis function kernels in SVMs. The number of feature functions equals the number of clusters L, that can be very high. Because of the thresholding the effective dimension is very low. Applying sparse vector routines allows for the efficient utilization of these feature functions.
It is well known in speech recognition that adding temporal acoustic context increases the accuracy of the acoustic model. Although the feature space X we used in our experiments already comprises acoustic context (see Section III), it turns out, that combining clustering features of subsequent time frames has a strong impact on the recognition performance.
Clustering features can be considered as a soft vector quantization. They are very flexible, efficiently to handle and in the following section we show that they provide good approximation abilities. However, the number of clusters and the context length has to be determined by experiments.

III. Experimental Results

All speech recognition experiments were performed on the WSJ0 corpus with a vocabulary of 5k words. Statistics for this corpus are given in Table I. Though its vocabulary size is small in comparison to more recent corpora, it is considered as a LVCSR task. The training corpus consists of 15 hours and the evaluation corpus of 0.4 hours of read speech. Since the official WSJ0 corpus does not provide a development set, 410 sentences were extracted from ten new speakers of the North American Business (NAB) task and used as a development set. The task has a closed vocabulary, that means all words in the development and evaluation corpus are known. All speech recognition systems were tuned on the development corpus and then applied to the evaluation corpus.
The acoustic front end of all experiments comprises 16 Mel-Frequency Cepstral Coefcient (MFCC) features. The MFCC features are normalized by a Vocal Tract Length Normalization (VTLN) and augmented with a voicedness feature. Feature vectors from nine consecutive frames are concatenated and a Linear Discriminative Analysis (LDA) is used to reduce the dimension to 33.
The GHMM baseline recognition system uses 1500 generalized triphone states, which were top down clustered using a decision tree, plus one silence state. The emission probabilities are modelled by Gaussian mixture distributions with a total of about 223k densities, all sharing a single diagonal covariance matrix.
For all recognitions a trigram language model has been used. The baseline system has been trained according to the ML criterion and has a word error rate (WER) of 3.57%. This is much better than the result reported in [13] (4.89 %) and slightly better then the result in [14] (3.72 %), where no VTLN is used.
All log-linear systems were initialized randomly and trained with the Rprop algorithm [15] until convergence of the convex

TABLE I
STATISTICS FOR THE WALL STREET JOURNAL 5K CORPUS (WSJ0)

	WSJ0		
	training	**dev.**	**eval.**
amount of acoustic data [h]	14.77	0.46	0.4
# sentences	7240	410	330
# words	130976	6784	5353

training criterion, which usually takes about 50 iterations. Note that because of the convexity of the optimization problem, the estimated parameters do not depend on the choice of the optimization algorithm. The priors $p(s_t)$ were set to the relative frequencies calculated from the alignment used in training. A further refinement of the estimation of the prior probabilities is to scale them with a positive factor, but we could not observe any improvements from this.

In order to train the model from scratch, we first assumed a linear time alignment and performed a realignment with the converged model. After five cycles of training and realignment we obtained a reasonable alignment, which has been used for all reported experiments.

In our experiments we observed that the performance of the system could be increased strongly by scaling the emission probabilities in (2) with a factor $\alpha > 0$, i.e. for recognition we replace (2) by

$$p(s_1^T, x_1^T) = \prod_{t=1}^{T} p(x_t|s_t)^{\alpha} p(s_t|s_{t-1}).$$

Notice that instead of scaling the mixture probability one could tune the language model scale and the time distortion penalties, but scaling the mixture probability is more convenient. Like all other scaling factors we tuned this scale on the development set. The estimated values for α range from 0.3 to 5.0.

Another essential heuristic in framewise training is to accumulate the silence state with lower weight, i.e. in equation (5) all weights are set to one, except for the silence weight, which is set to a small value. A good heuristic for the choice of this value is the relative frequency of the silence states in an existing alignment [8].

For recognition the HMM-recognizer can be used except for the calculation of the emission probabilities. Instead, the posterior probabilities $p(s_t|x_t)$ have to be computed. The denominator in (4) can be discarded in recognition, since it does not depend on the state. Hence, only the feature functions and the inner product of the feature functions and the parameters have to be computed. The complexity for these operations depends on the choice of feature functions. For the proposed feature functions the computation time is comparable to that of Gaussian Mixture likelihoods.

The quality of a speech recognition system correlates strongly with the number of model parameters, which depends on the number of HMM states and the number of feature functions. In Subsection III-A and III-B we focus on the choice of feature functions and keep the number of HMM states fixed to 130, corresponding to a three state HMM for every phoneme. In Subsection III-C we increase the number of states of the best systems by modeling context dependent phonemes.

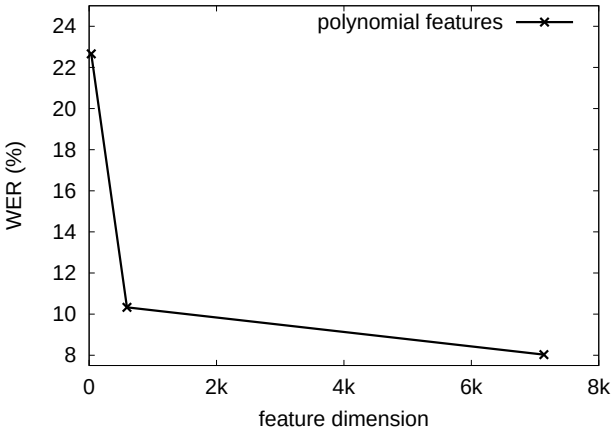

Fig. 1. WER for the WSJ0 corpus for polynomial features from order one up to three, corresponding to a feature dimension of 34, 595 and 7140.

A. Polynomial features

In the first set of experiments we evaluated polynomial features of order one, two and three. As shown in Figure 1 increasing the order of the model reduces the WER from 22.66% to 10.33% and 8.03%. Apart from being not feasible, experiments with order beyond three are not reasonable, because the experiments show that the impact of including higher order polynomials on the WER gets smaller, the higher the polynomial degree is.

The WER of the third order model is already quite low, considering that we trained a monophone system. On the other hand, training third order models is very slow. In order to improve the performance of the system one can increase the number of HMM states, but training time increases linearly with the number of states and therefore the training would be very time consuming.

B. Data dependent sparse features

The clustering features were constructed with the EM-algorithm for GMMs with tied diagonal covariance matrices. Note that the EM algorithm is applied to the acoustic training vectors without considering class labels, that means we do not rely on an existing GHMM system. Using tied covariance matrices yields more densities in dense regions at the expense of spending fewer densities for small clusters. In addition it prevents covariance matrices from being singular. For efficiency we applied a splitting procedure to obtain 2^{κ} densities, where κ denotes the number of splits. In the experiments we have used clustering features with κ equals to 7, 10 and 12. On average only 2.7 ($\kappa = 7$), 3.4 ($\kappa = 10$) and 4.9 ($\kappa = 12$) features were different from zero. The length of the symmetric window for the temporal acoustic context varied from 1 to 15. As a first step we determined the optimal window length. For

testing we used combined systems with first order polynomial and clustering features. We compared two types of systems. First we increased the context length up to 15 and kept the number of densities per time frame fixed. For comparison we kept the context length fixed to one and increased the number of splits. The results are shown in Figure 2. It turns out that in the beginning increasing the context length is more efficient than increasing the number of splits. After a context length of nine the error rate decreases only slightly. Consequently a context length of nine is used in all remaining experiments.

In the following we compare a system with clustering features only and combined systems with both polynomial and clustering features with varying number of splits. As expected the error rate decreases with the number of clusters (see Figure 3). In addition, including polynomial features reduces the error rate significantly. The best system with clustering features only achieves a WER of 7.06%. In combination with first order polynomial features an error rate of 6.58% and with second order polynomial features an error rate of 5.90% is obtained. This means including polynomial features decreases the error rate by 16.4% relative. This result is remarkable since polynomial features up to order two can be trained efficiently. Notice that the system with third order features achieves a WER of 8.03% (see Figure 1) while the system with 2^{10} clustering features only achieves 8.67% WER (see Figure 3), although it has more parameters. Nevertheless, due to the sparseness of the features, the training of the system with clustering features is more than 15 times faster compared to the system with third order features. Therefore the accuracy of this system can be improved easily by increasing the number of features or HMM states.

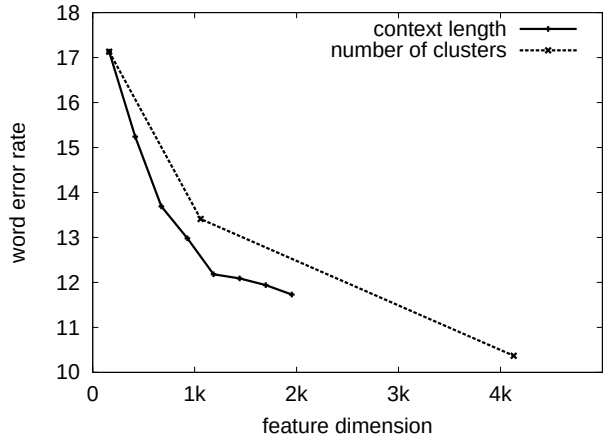

Fig. 2. WER for the WSJ0 corpus for first order models with clustering features created by 7 splits and a context length from 1,3,5,..,15 respectively with cluster features created by 7, 10 and 12 splits and fixed context length of 1.

C. The final system with combined features

Though the WER is already at a low level, the number of HMM states has to be increased to build a competitive system. With increasing the number of parameters models get more flexible, but due to possible overfitting the estimation of these parameters gets more difficult. This is the classic bias-variance-tradeoff problem. The baseline GMM system, which has about 7.6M parameters, provides an orientation which number of parameters are reasonable. Increasing the number of states for the combined log-linear system with 2^{12} clustering features to 1500, would give more than 56M parameters which is much higher than the number of parameters of the baseline system. Therefore we chose the log-linear models with first respectively second order polynomial features and 2^{10} clustering features for increasing the number of states. The results are plotted in Figure 4. Especially the second order model scales nicely with increased number of states. The combined first order model achieves a WER of 4.35%, and the combined second order model achieves a WER of 3.83%, which is already very close to the baseline system.

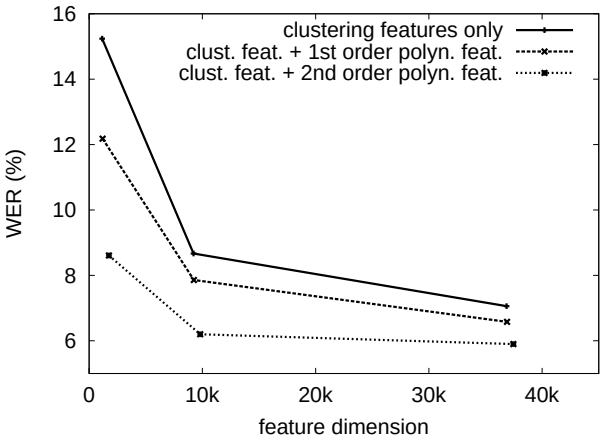

Fig. 3. WER for the WSJ0 corpus for systems with clustering features only, created by 7, 10 and 12 splits and a context length of 9, compared to systems with the same clustering features combined with first respectively second order polynomial features.

After a realignment and ten more training iterations the best system achieves a final WER of 3.55% (see Table II). The result is nearly exactly the same as the WER of the baseline system. Note that in comparison to other reported results, the baseline system is already highly competitive, but for a fair comparison the baseline system should be discriminatively trained. We know from other studies that this will cause a further reduction of the error rate of about 5% relative (see e.g. [14]) and want to investigate this in the future. However, the log-linear system is still in development and we expect further improvements, e.g. from switching to sentencewise training instead of framewise training.

IV. Discussion

We showed that log-linear models are competitive with GMMs for the application in continuous speech recognition. Moreover, since the log-linear speech recognition system is still experimental, further improvements of the system can be expected. We point out that a GMM as a starting point for the

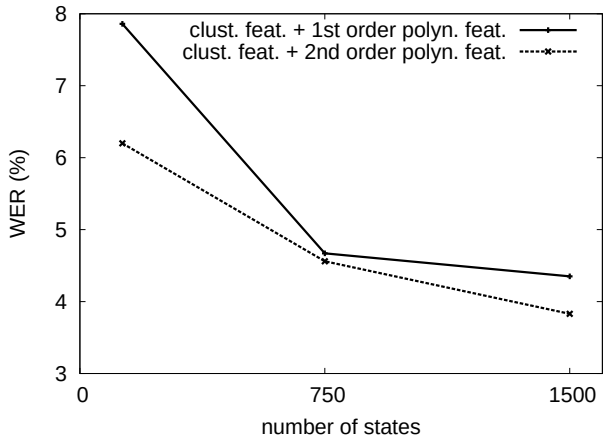

Fig. 4. WER for the WSJ0 corpus for first and second order models with clustering features created by 10 splits and a context length of 9 with increasing number of HMM states (130, 750, 1500)

TABLE II
WER FOR BASELINE AND LOG-LINEAR SYSTEMS ON THE WSJ0 CORPUS.

	polyn. feat.	**clust. feat.**		**WER (%)**
	order	# splits	ctxt. length	
GHMM baseline	-	-	-	3.57
log-lin. baseline	1	7	1	17.13
+ context	1	7	9	12.18
+ 2nd order	2	7	9	8.61
+ incr. # splits	2	10	9	6.20
+ 1500 states	2	10	9	3.83
+ realignment	2	10	9	3.55

log-linear training is not necessary.

From our experiments can be concluded that clustering features are superior to polynomial features. In [6] Hifny proposes very similar clustering features. In his experiments he uses a much higher number of densities (up to 130k compared to 9k in our best system). To cope with the resulting huge feature dimension he uses ℓ_1-regularization to obtain sparse models instead of ℓ_2-regularization as we do. Our experiments indicate that such a huge number of densities is not necessary for achieving competitive error rates. In addition we showed that combining clustering features with polynomial features leads to better results.

With the use of clustering features, log-linear models become as flexible as Gaussian Mixtures are. Another possibility to achieve this flexibility is to use log-linear models with hidden variables [11], but training these models is a non-convex problem. Doing the non-convex clustering before the convex log-linear training is a good compromise to obtain efficiency as well as flexibility of the model.

In the future we plan to evaluate the proposed model on larger corpora. With the usage of more refined training criteria, e.g. margin based criteria as proposed in [16], [8], we want to continue to improve the system. Furthermore, speaker adaptation techniques as CMLLR are an important part of state-of-the-art speech recognition systems and have to be included in the model. Finally, the utilization of more refined clustering procedures needs to be investigated.

ACKNOWLEDGMENT

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and also partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA.

REFERENCES

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[2] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP*, vol. 11, 1986, pp. 49–52.

[3] D. Povey and P. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP*, Orlando, FL, 2002, pp. 105–108.

[4] D. Kershaw, T. Robinson, and M. Hochberg, "Context-dependent classes in a hybrid recurrent network-HMM speech recognition system," *Advances in Neural Information Processing Systems, INIPS*, vol. 8, pp. 750–756, 1995.

[5] A. Ganapathisraju, "Support vector machines for speech recognition," Ph.D. dissertation, Mississippi State University, 2002.

[6] Y. Hifny and S. Renals, "Speech recognition using augmented conditional random fields," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 354–365, 2009.

[7] H.-K. J. Kuo and Y. Gao, "Maximum entropy direct models for speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 873–881, 2006.

[8] G. Heigold, D. Rybach, R. Schlüter, and H. Ney, "Investigations on convex optimization using log-linear HMMs for digit string recognition," in *Proc. Interspeech*, 2009.

[9] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proc. of the Seventeenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 591–598.

[10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. of the Eighteenth International Conference on Machine Learning, ICML*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[11] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Proc. Interspeech*, 2005, pp. 1117–1120.

[12] W. Rudin, *Principles of mathematical analysis*, 3rd ed. New York: McGraw-Hill Book Co., 1976, International Series in Pure and Applied Mathematics.

[13] Z.-J. Yan, B. Zhu, Y. Hu, and R.-H. Wang, "Minimum word classification error training of HMMs for automatic speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP*, 2008, pp. 4521–4524.

[14] W. Macherey, L. Haferkamp, R. Schlüter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," in *Proc. Interspeech*, Lisbon, Portugal, 2005, pp. 2133–2136.

[15] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The Rprop algorithm," in *Proc. of the IEEE International Conference on Neural Networks*, 1993, pp. 586–591.

[16] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney, "Modified MMI/MPE: A direct evaluation of the margin in speech recognition," in *Proc.of the 25th International Conference on Machine learning, ICML*. Helsinki, Finland: ACM, 2008, pp. 384–391.