On the Use of MLP Features for Broadcast News Transcription*

Petr Fousek, Lori Lamel, and Jean-Luc Gauvain

Spoken Language Processing Group, LIMSI-CNRS, France
{fousek,lamel,gauvain}@limsi.fr

Abstract. Multi-Layer Perceptron (MLP) features have recently been attracting growing interest for automatic speech recognition due to their complementarity with cepstral features. In this paper the use of MLP features is evaluated in a large vocabulary continuous speech recognition task, exploring different types of MLP features and their combination. Cepstral features and three types of Bottle-Neck MLP features were first evaluated without and with unsupervised model adaptation using models with the same number of parameters. When used with MLLR adaption on a broadcast news Arabic transcription task, Bottle-Neck MLP features perform as well as or even slightly better than a standard 39 PLP based front-end. This paper also explores different combination schemes (feature concatenations, cross adaptation, and hypothesis combination). Extending the feature vector by combining various feature sets led to a 9% relative word error rate reduction relative to the PLP baseline. Significant gains are also reported with both ROVER hypothesis combination and cross-model adaptation. Feature concatenation appears to be the most efficient combination method, providing the best gain with the lowest decoding cost.

1 Introduction

Over the last decade there has been growing interest in developing automatic speech-to-text transcription systems that can process broadcast data in a variety of languages. The availability of large text and audio corpora on the Internet has greatly facilitated the development of such systems, which nowadays can work quite well on unseen data that is similar to what has been used for training. However, there is still a lot of room for improvement for all system components including the acoustic front end, the acoustic, pronunciation and language models. One promising research direction is the use of MLP features in a large speech recognition task, in this case, the transcription of Arabic broadcast news from the DARPA GALE task.

Features for speech-to-text obtained from neural networks have recently been included as a component of a state-of-the-art LVCSR systems [1]. They are known to contain complementary information to cepstral features, which is why most often both features are used together.

* This work was in parts supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022 an in parts by OSEO under the Quaero program.

P. Sojka et al. (Eds.): TSD 2008, LNAI 5246, pp. 303–310, 2008.

Conventional neural network systems such as TANDEM [2] and TRAP [3] use three-layer MLPs trained to estimate phone posterior probabilities at every frame, which are then used as features for a GMM/HMM system. They are sometimes referred to as *probabilistic features*. The size of the MLP output features is reduced by a principal components analysis (PCA) transform. However, this might not necessarily be the optimal choice, especially when the dimensionality reduction is severe. The recently proposed *bottle-neck features* override this issue by employing four or five-layer MLPs and using outputs of a small hidden layer as features [4,5]. Not only does it allow for an arbitrary vector size, it also suggests using more MLP training targets for better discriminability.

Probabilistic features have never been shown to consistently outperform cepstral features in LVCSR. However, they can markedly improve the performance when used in conjunction with them. A number of multi-stream combination techniques have been successfully used for this purpose, four of which are studied in this work. These are MLP combination, feature concatenation, model adaptation and ROVER voting.

In this work, the bottle-neck architecture was used to deliver three types of MLP features, which differ in their input speech representations. Acoustic models are estimated using the three feature sets, and their performance is compared to a baseline system using PLP features. Different methods to combine the MLP and PLP features are explored, as well as combination of system outputs, with the goal of learning the most effective combination methods.

2 Arabic BN Task Description

The speech recognizer is a development version of the Arabic speech-to-text system component used in the AGILE participation in the GALE'07 evaluation. The transcription system has two main components, an audio partitioner and a word recognizer [6]. The audio partitioner is based on an audio stream mixture model, and serves to divide the continuous stream of acoustic data into homogeneous segments, associating cluster, gender and labels with each non-overlapping segment. The recognizer makes use of continuous density HMMs for acoustic modeling and n-gram statistics for language modeling. Each context-dependent phone model is a tied-state left-to-right CD-HMM with Gaussian mixture observation densities where the tied states are obtained.

Word recognition is performed in one or two passes, where each decoding pass generates a word lattice with cross-word, position-dependent, gender-independent acoustic models, followed by consensus decoding with 4-gram and pronunciation probabilities [6,7]. Unsupervised acoustic model adaptation is performed for each segment cluster using the CMLLR (Constrained Maximum Likelihood Linear Regression) and MLLR [8] techniques prior to second decoding pass.

A subset of the available Arabic broadcast news data was used to train acoustic models for the development system. This subset is comprised of 389 hours of manually transcribed data distributed by the Linguistic data consortium. These data were used to train the baseline gender-independent acoustic models, without maximum-likelihood linear transform (MLLT) or speaker-adaptive training (SAT). The models cover 30k contexts with 11.5k tied states, and have 32 Gaussians per state.

The language models were trained on corpora comprised of about 10 million words of audio transcriptions and 1 billion words of texts from a wide variety of sources. The recognition word list contains 200k non-vocalized, normalized entries. The language models result from the interpolation of models trained on subsets of the available data. The summed interpolation weights of the audio transcriptions is about 0.5. The pronunciation lexicon is represented with 72 symbols, including 30 simple consonants, 30 geminate consonants, 3 long and 3 short vowels, 3 vowels+tanwin, plus 3 pseudo phones for non-linguistic events (breath, filler, silence).

The test data is comprised of about 3 hours of broadcast news data referred to in the GALE community as the bnat06 development set. The out-of-vocabulary rate with this word list is about 2%, and the devset perplexity with a 4-gram language model is about 660.

3 MLP Features

Neural network feature extraction consists of two steps. The first step is *raw feature extraction* which constitutes the input layer to the MLP. Typically this vector covers a wide temporal context (100–500 ms) and therefore is highly dimensional. Second, the raw features are processed by the MLP followed by a PCA transform to yield the *HMM features*.

Two different sets of raw features are used, 9 frames of PLPs (9xPLP) and time-warped linear predictive TRAP (wLP-TRAP) [9]. The first set of raw features is based on the PLP features used in the baseline system which are mean and variance normalized per speaker. At each 10 ms frame, the MLP input is obtained by concatenating 9 successive frames of 13 PLP features (including energy) plus their first and second order derivatives (Δ and Δ^2), centered at the current frame. The feature vector has $9 \times 39 = 351$ values and covers a 150 ms window.

The second set of features is obtained by warping the temporal axis in the LP-TRAP feature calculation. Linear prediction is used to model the Hilbert envelopes of 500 ms long energy trajectories in auditory-like frequency sub-bands [10]. The input to the MLP are 25 LPC coefficients in 19 frequency bands, yielding $19 \times 25 = 475$ values which cover a 500 ms window. The naming conventions adopted for the various features sets are given in Table 1 along with how the raw features relate to the HMM features.

The bottle-neck architecture is based on a four layer MLP with an input layer, two hidden layers and an output layer. The second layer is large and it provides the necessary modeling power. The third layer is small, its size is equal to the required number of features. The output layer computes the estimates of the target class posteriors. Instead of using these posteriors as features, a PCA transform is applied to the outputs of the small hidden layer neurons (prior to a non-linear sigmoid function). A layer size of 39 was used in order to be able to more easily compare the performance of the MLP features to the PLP features.

Probabilistic MLPs are typically trained with phone targets. Since the size of the bottle-neck layer is independent of the number of output targets, it is quite easy to increase this number to improve the discrimination capacity of the MLP. Since there are often more differences between the states of the same phone than between different

Table 1. Naming conventions for MLP features and how the raw input features relate to the HMM features

ID	*Raw features (#)*	*HMM features (#)*
PLP	–	PLP+Δ+Δ^2 (39)
$\text{MLP}_{\text{9xPLP}}$	9x(PLP+Δ+Δ^2) (351)	MLP (39)
MLP_{wLP}	wLP-TRAP (475)	MLP (39)
MLP_{comb}	9x(PLP+Δ+Δ^2) + wLP-TRAP (826)	MLP (39)

states in the same position of different phones, it could be effective to replace the phone targets to by phone state targets. The phone state segmentations were obtained via a forced alignment using three-state triphone HMMs, with 69 phones and 3 non-linguistic units. The number of MLP targets was therefore increased from 72 to 210.

Since the MLP training is time-consuming, the MLP size and the amount of training data needs to be properly balanced. It is known that more data and/or more parameters always help, but at certain point the gain is not worth the effort. Table 2 gives the word error rate as a function of the amount of MLP training data. MLPs of a constant size (1.4 million parameters) were trained on various amounts of data using the 9xPLP raw features by gradually adding data from more speakers. HMMs were trained on the full 389 hour data set for all conditions.

The top part of the table gives WERs for phone targets. It can be seen that the improvement obtained by using additional data rapidly saturates with only a negligible gain when increasing the data by a factor of 10 (from 17 to 170 hours). The lower part of the table corresponds to using state targets. The change from phone targets to state targets brought a 2.4% relative reduction in WER (from 25.3% to 24.7%) with a MLP trained on a 17-hour data subset. The phone trained MLP correctly classified about 55% of unseen frames, whereas the state trained MLP was correct on about 50% of the frames. Given that the number of classes has tripled, it indicates that the state targets are indeed a good choice. In contrast to the phone targets, training state targets benefits from the additional data, with relative error rate reductions of 2-3% (24.2 to 23.4). The reference WER with the baseline PLP system, trained on the same data with the same model configuration, is 25.1%.

Since at the time of preparing this paper, training the MLP with wLP-TRAP features on the full corpus was not finished, the subsequent experiments were carried out using the MLP trained on 63 hours of speech recorded during the period from 2000 to 2002. Though not shown in the paper, partial experiments with the MLP trained on the full corpus show consistent improvements in performance over the values reported in the following sections.

4 System Combination

Experiments with system combinations were carried on using four types of features as listed in Table 1. The fourth feature set is obtained by combining the 9xPLP and the wLP-TRAP inputs to the MLP. All the four basic features were first evaluated without and with unsupervised acoustic model adaptation, as shown in the first four entries

Table 2. Word error rates (%) for phone and state based MLP as a function of the amount of training data (using 9xPLP raw features). All the HMMs are trained on the full 389 hours.

MLP targets	*MLP train data*	*WER (%)*
phones	1.5 hrs	27.3
	17 hrs	25.3
	170 hrs	25.0
states	17 hrs	24.7
	63 hrs	24.2
	301 hrs	23.4
PLP baseline		25.1

Table 3. Performance of PLP and MLP features, MLP combined features and feature concatenation

		WER (%)	
#	*Features*	*1-pass*	*2-pass*
1	PLP	25.1	22.5
2	$\mathrm{MLP}_{\mathrm{9xPLP}}$	24.2	22.7
3	$\mathrm{MLP}_{\mathrm{wLP}}$	25.8	23.1
4	$\mathrm{MLP}_{\mathrm{comb}}$	23.8	21.9
5	PLP + $\mathrm{MLP}_{\mathrm{9xPLP}}$	22.7	21.2
6	PLP + $\mathrm{MLP}_{\mathrm{wLP}}$	21.7	20.4
7	$\mathrm{MLP}_{\mathrm{9xPLP}}$ + $\mathrm{MLP}_{\mathrm{wLP}}$	22.2	21.0

in Table 3. The baseline performance of the standard PLP features with adaptation is 22.5%. Without adaptation, the $\mathrm{MLP}_{\mathrm{9xPLP}}$ features are seen to perform a little better (about 4% relative) than PLP, but with adaptation both $\mathrm{MLP}_{\mathrm{9xPLP}}$ and $\mathrm{MLP}_{\mathrm{wLP}}$ are slightly worse than PLP. This leads us to conclude that MLLR adaptation is less effective for MLP features than for PLP features. The $\mathrm{MLP}_{\mathrm{comb}}$ (the fourth entry in the table) is seen to perform better than PLP both with and without adaptation and suggests that combining raw features at the input to the MLP classifier is effective.

Next three means of fusing the information coming from the cepstral and the MLP features were evaluated. The simplest approach is to concatenate together the features at the input to the HMM system (this doubles the size of the feature vector, $2 \times 39 = 78$ features) and to train an acoustic model. Three possible pairwise feature concatenations were evaluated and the results are given in the lower part of Table 3. These concatenated features all substantially outperform the PLP baseline, by up to 9% relative, showing that feature concatenation is a very effective approach. Given the significantly better performance of the PLP + $\mathrm{MLP}_{\mathrm{wLP}}$ features over the PLP + $\mathrm{MLP}_{\mathrm{9xPLP}}$ and $\mathrm{MLP}_{\mathrm{9xPLP}}$ + $\mathrm{MLP}_{\mathrm{wLP}}$ features, the three-way concatenation was not tested as it was judged to be not worth the increased computational complexity needed to deal with the resulting feature vector size (3×39).

Two other more computationally expensive approaches were studied, cross model adaptation and ROVER [11]. Table 4 gives some combination results using cross

Table 4. Comparing cross-adaptation and ROVER for combining multiple systems

	WER (%)	
Combined systems	*1-pass*	*2-pass*
$3 \rightarrow 1$	25.8	21.5
$1 \rightarrow 3$	25.1	22.0
$7 \rightarrow 1$	22.2	20.7
$1 \rightarrow 7$	25.1	21.2
$1 \oplus 2 \oplus 3$	22.3	20.6
$1 \oplus 3$	23.3	21.0
$5 \oplus 6$	21.2	19.9
$1 \oplus 6 \oplus 7$	21.0	19.7

adaptation (top) and ROVER (bottom). The first entry is the result of adapting the PLP models with the hypotheses of the MLP_{wLP} system. The second entry corresponds to the reverse adaptation order, i.e. the MLP_{wLP} are adapted using the hypotheses of the PLP system. The next two entries use cross adaptation on top of feature concatenation. In the first 3 cases, cross adaptation reduces the WER (note that the 2nd pass error rates must be compared with those in Table 3). Larger gains are obtained when the PLP models are used in the second pass, supporting the earlier observation that MLLR adaptation is more effective for PLP features than for MLP features. This may be because the MLP already removes the variability due to the speaker or because other, perhaps non-linear, transformations are needed to adapt MLP features. The WERs in the bottom part of the table result from ROVER combination of the first or second pass hypotheses of the listed systems. ROVER combination of the three basic features performed better than the best pair-wise cross-adaptation amongst them ($3 \rightarrow 1$) however, neither combination outperformed the simple feature concatenation WER of 20.4% (entry 6 in Table 3). ROVER also helps when applied jointly with other combination methods (see the last two rows in Table 4), beating the baseline PLP system by up to 12% relative. This best ROVER result however requires 6 decoding passes!

It is interesting to observe that the PLP features are generally best combined with MLP_{wLP}, even though the $\text{MLP}_{\text{9xPLP}}$ gives better score than MLP_{wLP}. This may be due on one side to the fact that the $\text{MLP}_{\text{9xPLP}}$ features are derived from the PLPs, and on the other side that there is a larger difference in time spans between the standard PLP and the wLP-TRAP features.

Table 5. Best results after MLLR adaptation for different types of system combination of PLP and MLP_{wLP} features

Features	*WER (%)*	*Comment*
PLP + MLP_{wLP}	20.4	best feature concatenation
$1 \oplus 3$	21.0	best ROVER (1-3)
$\text{MLP}_{\text{wLP}} \rightarrow$ PLP	21.5	best cross-adaptation
MLP_{comb}	21.9	MLP combination

Table 5 summarizes the best results after adaptation obtained for each combination method with PLP and MLP_{wLP}. The systems are sorted by WER in ascending order. It appears that feature concatenation is a very efficient combination method, as it not only results in the lowest WER for 2 front-ends but it also has the lowest cost.

5 Summary

Three novel MLP feature sets derived using the bottle-neck MLP architecture have been evaluated in the context of an LVCSR system. One feature set is based on nine frames of PLP features and their derivatives, with a temporal span of 150 ms. The other feature set is an improved version of LP-TRAP and has a longer temporal span of 500 ms. Different schemes have been used to combine these two MLP feature sets with PLP features to determine the most effective approach.

Experiments were carried out on the Gale Arabic broadcast news task. When used with MLLR adaption, the MLP features perform as well or even slightly better than a standard PLP based front-end. Doubling the feature vector by combining the two feature sets led to a 9% relative WER reduction relative to the PLP baseline. Combining the same feature sets via cross-model adaptation or ROVER also gave improvement but to a lesser degree.

Feature concatenation appears to be the most efficient combination method, providing the best gain at the lowest decoding cost. In general, it seems best to combine features based on different time spans as they provide high complementarity.

It should be noted that as shown in the paper, the MLP system accuracy can be further improved by training the MLP on more data.

References

1. Zhu, Q., Stolcke, A., Chen, B.Y., Morgan, N.: Using MLP features in SRI's conversational speech recognition system. In: INTERSPEECH 2005, pp. 2141–2144 (2005)
2. Hermansky, H., Ellis, D., Sharma, S.: TANDEM connectionist feature extraction for conventional HMM systems. In: ICASSP 2000, Istanbul, Turkey (2000)
3. Hermansky, H., Sharma, S.: TRAPs - classifiers of TempoRAl Patterns. In: ICSLP 1998 (November 1998)
4. Grézl, F., Karafiát, M., Kontár, S., Černocký, J.: Probabilistic and bottle-neck features for LVCSR of meetings. In: ICASSP 2007, April 2007, pp. 757–760. IEEE Signal Processing Society, Hononulu (2007)
5. Grézl, F., Fousek, P.: Optimizing bottle-neck features for LVCSR. In: ICASSP 2008, Las Vegas, ND (2008)
6. Gauvain, J., Lamel, L., Adda, G.: The LIMSI Broadcast News Transcription System. Speech Communication 37(1-2), 89–108 (2002)
7. Lamel, L., Messaoudi, A., J.L.G.: Improved Acoustic Modeling for Transcribing Arabic Broadcast Data. In: Interspeech 2007, Antwerp, Belgium (2007)
8. Leggetter, C., Woodland, P.: Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Computer Speech and Language 9(2), 171–185 (1995)

9. Fousek, P.: Extraction of Features for Automatic Recognition of Speech Based on Spectral Dynamics. PhD thesis, Czech Technical University in Prague, Faculty of Electrical Engineering, Prague (March 2007)
10. Athineos, M., Hermansky, H., Ellis, D.P.: LP-TRAP: Linear predictive temporal patterns. In: ICSLP 2004 (2004)
11. Fiscus, J.: A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER) (1997)