THE PHILIPS/RWTH SYSTEM FOR TRANSCRIPTION OF BROADCAST NEWS

P. Beyerlein (1), X. Aubert (1), R. Haeb-Umbach (1), M. Harris (1), D. Klakow (1), A. Wendemuth (1), S. Molau (2), Michael Pitz (2), A. Sixtus (2)

(1) Philips Research Laboratories, Weisshausstrasse 2, D-52066 Aachen, Germany

(2) Lehrstuhl f. Informatik VI, Aachen University of Technology, D-52056 Aachen, Germany

ABSTRACT

The paper contains a description of the Philips/RWTH 1998 HUB4 system which was build in a joint effort of Philips Research Laboratories Aachen and Aachen University of Technology. We will focus our discussion on recent improvements compared to the original 1997 HUB4 system [Beyerlein$^+$ 1998] and evaluate them on the HUB4'97 evaluation data. The paper will deal with

1. a rough system overview, including slight modifications in feature extraction, training and recognition
2. improvements obtained in audio stream segmentation and segment clustering,
3. log-linear interpolation of "distance" language models,
4. and the integration of various acoustic and language models via Discriminative Model Combination (DMC).

The performance of the described system is 23% (relative) better than the performance of the 1997 Philips HUB4 system. A word error rate of 17.9% was achieved on the 1997 HUB4 evaluation set, compared to 23.5% using the original 1997 system.

1. System Overview

1.1. Feature Extraction

In the acoustic front end, mel-frequency cepstral coefficients were computed. A feature vector consists of 15 static features, 15 linear regression delta features, the frame energy and its first- and second-order derivatives, resulting in a 33-component feature vector. Three consecutive feature vectors were concatenated into a 99-component vector to which a linear discriminant analysis (LDA) was applied. The gender-independent LDA matrix has been estimated on the Broadcast News (BN) training data. The final feature vector consisted of the 35 vector components with the largest eigenvalues. Vocal tract normalization (VTN) [Haeb$^+$ 1998] was applied in recognition only. The hypothesized transcription required by VTN had been obtained from a first trigram decoding without VTN. The cepstral features were normalized per segment by cepstral mean subtraction and by unit variance normalization.

1.2. Acoustic Training

We trained gender-dependent F-condition-independent models on 96 hours of the acoustic BN training data. The acoustic context was modeled by word-internal triphone models, cross-word triphone models and word-internal pentaphone models, where phrases of frequently spoken words were treated as a single word [Beyerlein$^+$ 1998]. In the acoustic modelling we employed continuous mixtures of Laplacian densities with a single, globally pooled deviation vector. Decision tree clustering was applied for a robust within-word, cross-word and pentaphone modeling.

1.3. Decoding

The decoder used a time-synchronous search algorithm based on the concept of word-dependent tree copies and integrates the trigram language-model constraints in one single pass. The pruning strategy included a look-ahead technique of the language model probabilities [Ortmanns$^+$ 1998]. Recently the search algorithm was extended to allow a one-pass trigram cross-word triphone decoding. The best sentence hypothesis is produced as well as a word lattice, both being used in the subsequent decoding stages performing acoustic adaptation and DMC. Decoding was done in a number of stages:

- First a trigram decoding using within-word triphone models was carried out. The resulting hypothesized word sequence was used for VTN and MLLR adaptation [Beyerlein$^+$ 1998].
- Using the adapted models the trigram decoding was repeated, producing lattices as output followed by DMC [Beyerlein 1997].

2. Audio Stream Segmentation

When applying automatic speech recognition to Broadcast News data, a preliminary segmentation step is required. The goal of this pre-processing stage is to partition the whole audio stream into reasonably short segments while discarding the non-linguistic portions. Sim-

ilar speaker segments are then clustered together, allowing for robust adaptation.
The segmentation used in the 1998 evalution was as follows:

- Non-speech passages were eliminated using a Gaussian Mixture Model (GMM) decoder that recognizes speech and non-speech.
- Subsequently, the passages of speech are divided at changes in speaker or background conditions using the Bayesian Information Criterion (BIC) as described in [Chen+ 1998].

The segmentation used in the 1997 HUB4 evaluation was based on using gender-dependent phone decoders (PHONE-DEC.) with additional non-speech units (see [Beyerlein+ 1998]).
For a comparison of these two segmentations with the ideal NIST-PE segmentation, the following quantities were measured:

- The amount of lost speech (misclassified as noise).
- The percentage of frames belonging to the dominant speaker in each segment, called the segment purity.
- The average segment length.
- The number of words that are cut at misplaced segment boundaries.

approach	speech lost (%)	purity (%)	avg. segment length	words cut (%)
PHONE-DEC.	0.35	97.6	7.33s	0.29
GMM/BIC	0.26	97.7	18.86s	0.23
NIST-PE	0	100	15.87	0

Table 1: *Comparison of segmentation approaches on HUB4'97 evaluation test set*

Table 1 summarizes the segmenter quality for the two described approaches and for the official NIST-PE segmentation. The GMM+BIC segmenter results in segments of longer length, whereas its purity is similar to that of the phone decoder.
Speaker clusters were created using a new bottom-up segment clustering algorithm based on the symmetric Kullback-Leibler distance and an additional neighbourhood penalty favouring clustering of neighbouring segments. Table 2 compares the simplified nearest neighbour clustering algorithm (SNN) used in the 1997 evaluation [Beyerlein+ 1998], with the new bottom-up approach and with an ideal clustering, where all segments

approach (segmentation +clustering)	cluster purity (%)	WER before VTN/MLLR (%)	WER after VTN/MLLR (%)
PHONE-DEC. + SNN	73.6	23.7	22.6
GMM/BIC + bottom-up	89.1	23.4	21.0
NIST-PE +ideal cl.	100	21.8	20.0

Table 2: *Word error rates (%) on HUB4'97 evaluation test set for different segmentation and clustering methods using a one-pass trigram decoding*

of one and the same speaker are clustered together. The bottom-up clustering method resulted in a higher purity than the nearest neighbour method, leading to reduced word error rates after adaption.

3. Log-Linear Interpolation of Language Models

3.1. Log-Linear interpolation

In [Klakow 1998] we suggested a new language modeling method called log-linear interpolation (LLI) which is related to maximum entropy models but has all the flexibility and the same number of free parameters as linear interpolation. Log-Linear interpolation is defined by

$$p_\Lambda(w|h) = \frac{1}{Z_\Lambda(h)} \prod_i p_i(w|h)^{\lambda_i} \qquad (1)$$

where $p_i(w|h)$ are the models to be combined and λ_i the weights. $Z_\Lambda(h)$ is the normalization constant. This model (1) can be interpreted as a linear interpolation of scores. However, an additional score from the normalization $Z_\Lambda(h)$ has to be added. We decided to optimize the log-likelihood

$$F(\Lambda) = \sum_{h\,w} f(h\,w) \log\left(\frac{1}{Z_\Lambda(h)} \prod_i p_i(w|h)^{\lambda_i}\right) \qquad (2)$$

with respect to the λ_i. Here, $f(h\,w)$ are the frequencies of the M-gram 'hw' in the cross-validation set. In table 3 the perplexities on the 1997 evaluation data are summarized. All models in this table are trained on BN. As a reference, the bigram, trigram and fourgram perplexity are also given. Firstly, a nice improvement can be achieved by combining a bigram and a distance-1 bigram using LLI. This model has a trigram-context but the full trigram is still better. When the same experiment is performed for a fourgram context, the situation changes. Now the LLI-combined model based on the trigram and distance-{0,1,2} bigrams is better than the

Model	PP
Bigram (=d0)	216
Trigram	150
Fourgram	144
Lin d0 + d1	204
LLI d0 + d1	175
Lin Tri +d0 +d1 +d2	146
LLI Tri +d0 +d1 +d2	136
Lin Tri +d0 ...+d5	146
LLI Tri +d0 ...+d5	130

Table 3: *Perplexities for log-linear interpolation of language models on the HUB4'97 evaluation set*

full fourgram. Because of memory restrictions, we did not train a backing-off sevengram. However, building the corresponding model following the pattern just described gives an additional improvement. Note also, that linear interpolation (denoted by 'Lin' in the table) as a method of combination is not competitive.

3.2. Optimized Distance Models

We are left with the problem of improving the performance of the models to be combined by LLI. This will now be illustrated for the distance-2 bigram. We trained initial distance-2 bigrams on BN. Those bigrams were then used to train classes. Note that this gives classes different from the standard bigram classes. Based on this classification a distance-2 class bigram is trained. In addition, a separate distance-2 bigram is constructed from the North American News Text Corpus. All models are combined by linear interpolation. This optimization scheme was used to build all component models, which were then combined by LLI. The first row in Table 4 gives the perplexity for the distance-2 bigram trained on BN only and the second row the optimized distance-2 bigram (denoted by 'Opt' in the table). The last two rows of the table compare the LLI combination of the component models trained on BN only with the optimized component models. Perplexity is reduced by 15%.

Model	PP
d2 Bigram BN	739
d2 Bigram Opt	661
LLI Tri +d0 +d1 +d2 BN	136
LLI Tri +d0 +d1 +d2 Opt	118

Table 4: *Perplexities for log-linear interpolation with an optimized distance-2 bigram model on the HUB4'97 evaluation set*

4. DMC

Discriminative model combination [Beyerlein 1997] aims at an optimal integration of all given (acoustic and language) models into one log-linear posterior probability distribution. Let us assume that we are given M different acoustic and language models, which are identified by numbers $j = 1, \ldots, M$. From model j we can compute the posterior probability $p_j(k|x)$ of a hypothesized class k given an observation x. These models are now log-linearly combined into a distribution of the exponential family:

$$p_{\Lambda}(k|x) \quad = \quad e^{-\log Z_{\Lambda}(x) + \sum_{j=1}^{M} \lambda_j \log p_j(k|x)} \tag{3}$$

The coefficients $\Lambda = (\lambda_1, ..., \lambda_M)^T$ can be interpreted as weights of the models j within the model combination (3). The value $Z_{\Lambda}(x)$ is a normalization constant. As opposed to the maximum entropy approach, which leads to a distribution of the same functional form, the coefficients Λ are optimized with respect to the decision error rate of the discriminant function (4):

$$\log \frac{p_{\Lambda}(k|x)}{p_{\Lambda}(k'|x)} \quad = \quad \sum_{j=1}^{M} \lambda_j \log \frac{p_j(k|x)}{p_j(k'|x)} \tag{4}$$

This approach is called "Discriminative Model Combination". If only one acoustic and one language model are combined, DMC will optimize the so called language weight (or language model factor). DMC allows for the integration of any model into an optimal decoder, since the weight λ_j of the model j within the combination depends on its ability to provide information for correct classification.

4.1. DMC Training

So far DMC was used to optimize a large vocabulary continuous speech recognition (LVCSR) system at the model level, although it could be applied to other problems in pattern recognition due to its general formulation. In LVCSR systems the spoken utterance is used as observation x and any hypothesized sentence can be regarded as class k. For DMC training we are given a set of sentences $n = 1, \ldots, N$. For each of the training sentences we know the observation x_n (spoken utterance) and the correct class assigment k_n (spoken word sequence). Using a preliminary decoding (if appropriate) we can define the set of rival classes $k \neq k_n$ and we can compute the number of word errors of the rival class k with the help of the Levenshtein distance $\mathcal{L}(k_n, k)$. The model combination should then minimize the word error count $E(\Lambda)$:

$$E(\Lambda) \quad = \quad \sum_{n=1}^{N} \mathcal{L}\left(k_n, \arg\max_{k \neq k_n} \left(\log \frac{p_{\Lambda}(k|x_n)}{p_{\Lambda}(k_n|x_n)}\right)\right)$$

on representative training data to assure optimality on an independent test set. Since this optimization criterion is not differentiable we approximate it in analogy to the well-known MCE training by a smoothed word error count:

$$E_{MWE}(\Lambda) = \sum_{n=1}^{N} \sum_{k \neq k_n} \mathcal{L}(k, k_n) S(k, n, \Lambda), \quad (5)$$

where $S(k, n, \Lambda)$ is a smoothed indicator function. $S(k, n, \Lambda)$ should be close to one if the classifier (4) will select hypothesis k and it should be close to zero if the classifier (4) will reject hypothesis k. One possible indicator function with these properties is

$$S(k, n, \Lambda) = \frac{\left(\frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)}\right)^\eta}{\sum_{k'} \left(\frac{p_\Lambda(k'|x_n)}{p_\Lambda(k_n|x_n)}\right)^\eta}, \quad (6)$$

where η is a suitable constant. Optimization of $E_{MWE}(\Lambda)$ with respect to Λ leads to an iterative gradient descent scheme. Another possible indicator function with similar properties is the following 2-nd degree function:

$$S(k, n, \Lambda) = \begin{cases} \left(\frac{g+B}{A+B}\right)^2 & , \; -B < g < A \\ 0 & , \; g > A \\ 0 & , \; g < -B \end{cases} \quad (7)$$

with

$$g = \log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)},$$

which gives a closed form matrix solution for Λ. The values A, B determine the form of the 2-nd degree function and the set of hypotheses used for the training. Both indicator functions lead to similar and reasonable DMC coefficients λ_j.

4.2. DMC in the HUB4 System

The training of the DMC coefficients was carried out on lattices of the HUB4 development data. The lattices, which were obtained by the one-pass trigram decoding (section 1.3.), were expanded and rescored using the following phrase-based acoustic (section 1.2.) and language (section 3.) models:

- VTN/MLLR adapted word-internal triphones (wwad)
- VTN/MLLR adapted cross-word triphones (xwad)
- VTN/MLLR adapted word-internal 5-phones (5wwad)
- Unigram, Bigram, Trigram, d1 Bigram (tgset)
- Unigram, Bigram, Trigram, (tgset2)
- tgset, d2 Bigram (fgset).

The obtained scores were interpolated using DMC resulting in the final system output. Table 5 gives an overview over several decodings. In a first decoding iteration a system capturing a phrase-based cross-word pentaphone context and a trigram language model context was built ($wwad + xwad + 5wwad + tgset$). This system shows a word error rate of 18.9% compared to the baseline error rate of 20.7%. In a second decoding iteration ($*$), the adaptation of the acoustic and language models was repeated based on the output of the $wwad + xwad + 5wwad + tgset$ system. The system was extended to a fourgram context by adding the d2-Bigram language model to the combined set of models. Note that the weights of the log-linear language models interpolation described in section 3. are similar to the weights obtained from DMC ! The $wwad + xwad + 5wwad + fgset^*$ system showed an word error rate of 17.9% on the HUB4'97 evaluation data. The log-linear interpolation of acoustic and language models via DMC seems to be more powerful than a simple voting at the level of the recognized word sequence as is done with ROVER [Fiscus 1997]. Table 6 shows the obtained results. For the tests the NIST SCTK-1.2 ROVER software was used.

models	M	WER
xwad+tg (Baseline)	2	20.7
wwad+xwad+tg	3	20.2
wwad+xwad+5wwad+tg	4	19.5
wwad+xwad+5wwad+tgset	7	18.9
wwad+xwad+5wwad+fgset*	8	17.9

Table 5: *Word error rates (%) for the log-linear combination of acoustic and language models using DMC on the HUB4'97 evaluation data*

models	DMC (#models)	ROVER (#systems)
wwad+tg	21.6 (2)	- (1)
xwad+tg	20.7 (2)	- (1)
wwad+xwad+tg	20.2 (3)	22.5 (2)
wwad+xwad+5wwad+tg	19.5 (4)	19.9 (3)
wwad+xwad+5wwad+tgset2	19.5 (6)	20.0 (9)
wwad+xwad+5wwad+tgset	18.9 (7)	20.2 (12)

Table 6: *Comparism of ROVER and DMC on the HUB4'97 evaluation data*

5. SUMMARY

The key features of the Philips/RWTH HUB4 system were described. Due to a better segmentation algorithm, the reduction of search errors using a one-pass trigram decoding, improved language models and more acoustic and language model training data the word error rate of the system could be reduced from 23.5% to 20.7% on the HUB4'97 evaluation data. With help of two DMC iterations, several adapted acoustic and language models with longer context could be exploited properly, which reduced the error rate from 20.7% to 17.9%. In the 1998 HUB4 evaluation word error rates of 18.5% on 'File1' and of 16.8% on 'File2' were reported for the described system.

References

[Beyerlein 1997] P. Beyerlein, "Discriminative Model Combination", in Proc. 1997 IEEE ASRU Workshop, Santa Barbara, pp. 238-245, Dec. 1997.

[Beyerlein+ 1998] P. Beyerlein, X. Aubert, R. Haeb-Umbach, D. Klakow, M. Ullrich, A. Wendemuth and P. Wilcox, "Automatic Transcription of English Broadcast News". Proc. of the DARPA Broadcast News Transcription and Understanding Workshop, Virginia, February 1998.

[Chen+ 1998] S.S. Chen, P.S. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion", in Proc. DARPA Broadcast News Transcription and Understanding Workshop, VA, Feb. 1998.

[Fiscus 1997] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)" in Proc. 1997 IEEE ASRU Workshop, Santa Barbara, pp. 347-354, Dec. 1997.

[Haeb+ 1998] R. Haeb-Umbach, X. Aubert, P. Beyerlein, D. Klakow, M. Ullrich, A. Wendemuth and P. Wilcox, "Acoustic Modeling in the Philips Hub-4 Continuous-Speech Recognition System". Proc. of the DARPA Broadcast News Transcription and Understanding Workshop, Virginia, February 1998.

[Klakow 1998] D. Klakow, "Log-Linear Interpolation of Language Models", in Proc. ICSLP'98, 1695-1698, Sidney, November 1998

[Ortmanns+ 1998] S. Ortmanns, A. Eiden, H. Ney, "Improved Lexical Tree Search for Large Vocabulary Speech Recognition", Proc. of ICASSP'98, pp. 817-820, Seattle, May 1998.