The LIMSI 1998 Hub-4E Transcription System

Jean-Luc Gauvain, Lori Lamel, Gilles Adda, Michèle Jardino

Spoken Language Processing Group
LIMSI-CNRS, BP 133
91403 Orsay cedex, FRANCE
{gauvain,lamel,gadda,jardino}@limsi.fr

ABSTRACT

In this paper we report on our Nov98 Hub-4E system, which is an extension of our Nov97 system[4]. The LIMSI system for the November 1998 Hub-4E evaluation is a continuous mixture density, tied-state cross-word context-dependent HMM system. The acoustic models were trained on the 1995, 1996 and 1997 official Hub-4E training data containing about 150 hours of transcribed speech material. 65K word language models were obtained by interpolation of backoff n-gram language models trained on different text data sets. Prior to word decoding a maximum likelihood partitioning algorithm segments the data into homogenous regions and assigns gender, bandwidth and cluster labels to the speech segments. Word decoding is carried out in three steps, integrating cluster-based MLLR acoustic model adaptation. The final decoding step uses a 4-gram language model interpolated with a category trigram model.

The main differences compared to last year's system arise from the use of additional acoustic and language model training data, the use of divisive decision tree clustering instead of agglomerative clustering for state-tying, generation graph word using adapted acoustic models, the use of interpolated LMs trained on different data sets instead of training a single model on weighted texts, and a 4-gram LM interpolated with a category model. The overall word transcription error on the Nov98 evaluation test data was 13.6%.

INTRODUCTION

In this paper we describe our Nov98 broadcast news transcription system and report on our development work prior to the Nov98 Hub4 evaluation test. The goal of the DARPA Hub-4 task is to transcribe radio and television news broadcasts. Radio and television broadcasts contain signal segments of various linguistic and acoustic natures, with abrupt or gradual transitions between segments. Data partitioning serves to divide the continuous stream of acoustic data into homegenous segments, associating appropriate labels with the segments. The segmentation and labeling procedure[3] first detects and rejects non-speech segments, and then applies an iterative maximum likelihood segmentation/clustering procedure to the speech segments. The result of the partitioning process is a set of speech segments with cluster, gender and telephone/wideband labels.

The LIMSI Nov98 speech recognizer uses continuous density HMMs with Gaussian mixture for acoustic modeling and *n*-gram statistics estimated on large text corpora 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 by means of a decision tree. Word recognition is performed in three steps: 1) initial hypothesis generation, 2) word graph generation, 3) final hypothesis generation, each with two passes. The initial hypothesis are used in cluster-based acoustic model adaptation using the MLLR technique[10] prior to word graph generation and in all subsequent decoding passes. The final hypothesis is generated using a 4-gram interpolated with a category trigram model with 270 automatically generated word classes[8].

The acoustic models were trained on about 150 hours of Broadcast News data. Language models (LMs) were obtained by interpolation of backoff n-gram language models trained on different data sets: BN transcriptions, NAB newspapers and AP Wordstream texts prior to Sep95 and after July96, and transcriptions of the BN acoustic data. The recognition vocabulary contains 65K words and has a lexical coverage of over 99% on all evaluation test sets. A pronunciation graph is associated with each word, represented using a set of 48 phones.

Our development work was aimed at improving the partitioning algorithm[4, 5] and improving the acoustic and language models. The main differences relative to our Nov97 system are the use of additional acoustic and language model training data, the use of divisive decision tree clustering instead of agglomerative clustering for state-tying, the generation of word graphs using adapted acoustic models as well as acoustic model adaptation prior to successive decoding passes, the use of interpolated LMs trained on different data sets instead of training a single model on weighted texts, and a 4-gram LM interpolated with a category model.

In the remainder of this paper we provide an overview of the LIMSI Nov98 Hub-4E system, summarizing some of our development work in preparation for the Nov98 Hub-4E evaluation, and differences with our Nov97 system.

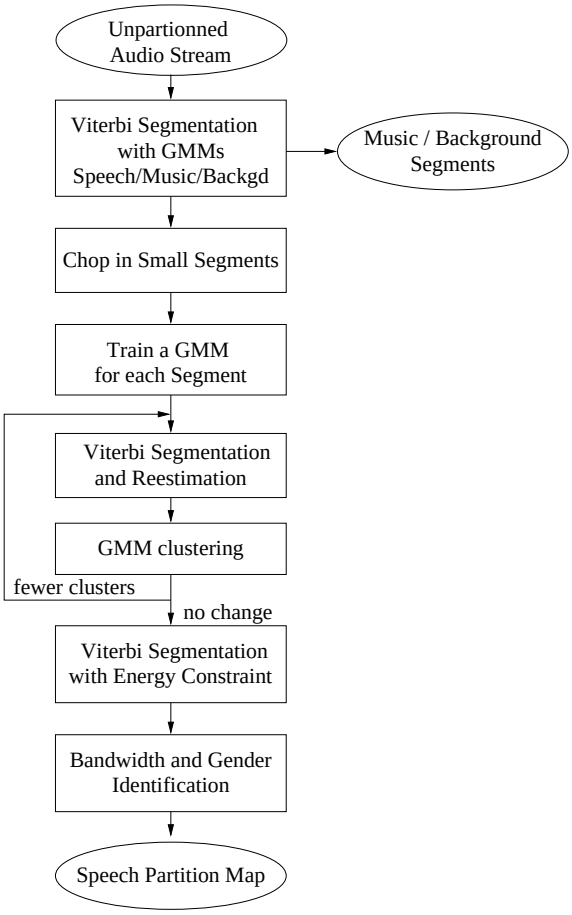

Figure 1: Partitioning algorithm.

DATA PARTITIONING

While it is evidently possible to transcribe the continuous stream of audio data without any prior segmentation, partitioning offers several advantages over this straight-foward solution. First, in addition to the transcription of what was said, other interesting information can be extracted such as the division into speaker turns and the speaker identities. Prior segmentation can avoid problems caused by linguistic discontinuity at speaker changes. By using acoustic models trained on particular acoustic conditions, overall performance can be significantly improved, particularly when cluster-based adaptation is performed. Finally by eliminating non-speech segments and dividing the data into shorter segments (which can still be several minutes long), reduces the computation time and simplifies decoding.

The segmentation and labeling procedure introduced in [4] is shown in Figure 1. First, the non-speech segments are detected (and rejected) using Gaussian mixture models (GMMs). The GMMs each with 64 Gaussians serve to detect speech, pure-music and other (background). The acoustic feature vector used for segmentation contains 38 parameters. It is the same as the recognition feature vector except that it does not include the energy, although the delta energy parameters are included. The GMMs were each trained on about 1h of acoustic data, extracted from the training data after segmentation with the transcriptions. The speech model was trained on data of all types, with the exception of pure music segments and silence portions of segments transcribed as speech over music. In order to detect speech in noisy conditions a second speech GMM was trained on the F4 segments in the 1996 data set. These model are expected to match all speech segments. The music model was trained only on portions of the data that were labeled as pure music, so as to avoid mistakenly detecting speech over music segments. The silence model was trained on the segments labeled as silence during forced alignment, after excluding silences in segments labeled as containing speech in the presence of background music. All test segments labeled as music or silence are removed prior to further processing.

A maximum likelihood segmentation/clustering iterative procedure is then applied to the speech segments using GMMs and an agglomerative clustering algorithm. Given the sequence of cepstral vectors corresponding to a show $(x_1, \ldots, x_T)$, the goal is to find the number of sources of homogeneous data (modeled by the p.d.f. $f(\cdot|\lambda_k)$ with a known number of parameters) and the places of source changes. The result of the procedure is a sequence of non-overlaping segments $(s_1, \ldots, s_N)$ with their associated segment cluster labels $(c_1, \ldots, c_N)$, where $c_i \in [1, K]$ and $K \leq N$. Each segment cluster is assumed to represent one speaker in a particular acoustic environment. In absence of any prior knowledge about the stochastic process governing (K, N) and the segment lengths, we use as objective function a penalized log-likelihood of the form

$$\sum_{i=1}^{N} \log f(s_i|\lambda_{c_i}) - \alpha N - \beta K$$

where $\alpha > 0$ and $\beta > 0$. The terms αN and βK, which can be seen as segment and cluster penalties, correspond to the parameters of exponential prior distributions for N and K. It is easy to prove that starting with overestimates of N and K, alternate Viterbi reestimation and agglomerative clustering gives a sequence of estimates of (K, N, λ_k) with non decreasing values of the objective function. In the Viterbi step we reestimate (N, λ_k) so as to increase $\sum_i \log f(s_i|\lambda_{c_i}) - \alpha N$ (i.e. adding a segment penalty α in the Viterbi search) whereas in the clustering step two or more clusters can be merged as long as the resulting log-likelihood loss per merge is less than β.[1] Since merging two models can reduce the number of segments, the change in segment penalty is taken into account during clustering. This algorithm stops when no merge is possible. A constraint on the cluster size is used to ensure that each cluster corre-

[1] This clustering criterion is closely related to the MDL or BIC criterion.

sponds to at least 10s of speech. (Recall that the previously rejected non-speech segments are not considered here.)

For single Gaussian models the merging criterion is easy to implement since the log-likelihood loss can be directly computed from the sufficient statistics of the corresponding segments[6, 9]. In the more general case of Gaussian mixtures, there are no sufficient statistics and there is no direct solution to compute the resulting mixture and/or the log-likelihood loss. We can envison estimating the new mixture from the data but this is a costly procedure. Another solution that we adopted for this work is to modify the objective function, replacing the likelihood function by the complete data likelihood of the Gaussian mixtures and extending the maximum likelihood clustering method to the Gaussian level. To estimate the log-likelihood loss for two Gaussian mixtures, we simply have to compute the sum of the log-likelihood loss while clustering the Gaussians of the 2 mixtures (until we get the desired number of Gaussians). We have used 8 mixture components per cluster, so to compute the log-likelihood loss induced by merging two clusters agglomerative clustering is performed starting with 16 Gaussians until 8 Gaussians are left.

The process is initialized using a simple segmentation algorithm based on the detection of spectral change (similar to the first step used in the CMU'96 system[11]). The threshold is set so as to over-generate segments, roughly 5 times as many segments as true speaker turns. Initially, the cluster set consists of a cluster per segment. This is followed by Viterbi training of the set of GMMs (one 8-component GMM per cluster). This procedure is controlled by 3 parameters: the minimum cluster size (10s), the maximum log-likelihood loss for a merge (α), and the segment boundary penalty (β). When no more merges are possible, the segment boundaries are refined using the last set of GMMs and an additional relative energy-based boundary penalty, within a 1s interval. This is done to locate the segment boundaries at silence portions, attempting to avoid cutting words (but sometimes this still occurs).

Speaker-independent GMMs corresponding to wideband speech and telephone speech (each with 64 Gaussians) are then used to label telephone segments. This is followed by segment-based gender identification, using 2 sets of GMMs with 64 Gaussians (one for each bandwidth). The result of the partitioning process is a set of speech segments with cluster, gender and telephone/wideband labels.

In developing the partitioner we used the dev96 data set, and we evaluated the frame level segmentation error (similar to [7]) on the 4 half-hour shows in the eval96 test data using the manual segmentation found in the reference transcriptions. The NIST transcriptions of the test data contain segments that were not scored, since they contain overlapping or foreign speech, and occasionally there are small gaps between consecutive transcribed segments. Since we considered that the partitioner should also work correctly on these portions, we relabeled all excluded segments as speech, music or other background.

Show	*1*	*2*	*3*	*4*	*Avg*
Frame Error	7.9	2.3	3.3	2.3	3.7
M/F Error	0.4	0.6	0.6	2.2	1.0
#spkrs/#clusters	7/10	13/17	15/21	20/21	-
ClusterPurity	99.5	93.2	96.9	94.9	95.9
Coverage	87.6	71.0	78.0	81.1	78.7

Table 1: Top: Speech/non-speech frame segmentation error (%), using NIST labels, where missing and excluded segments were manually labeled as speech or non-speech. Bottom: Cluster purity and best cluster coverage (%).

Table 1(top) shows the segmentation frame error rate and speech/non-speech errors for the 4 shows. The average frame error is 3.7%, but is much higher for show 1 than for the others. This is due to a long and very noisy segment that was deleted. Averaged across shows the gender labeling has a 1% frame error. In addition to these errors, there are 6.2% female speech frames deleted (largely due to the same segment) and 1.7% of the male frames deleted. The bottom of Table 1 shows measures of the cluster homogeneity. The first entry gives the total number of speakers and identified clusters per file. In general there are more clusters than speakers, as a cluster can represent a speaker in a given acoustic environment. The second measure is the cluster purity, defined as the percentage of frames in the given cluster associated with the most represented speaker in the cluster. (A similar measure was proposed in [1], but at the segment level.) The table shows the weighted average cluster purities for the 4 shows. On average 96% of the data in a cluster comes from a single speaker. When clusters are impure, they tend to include speakers with similar acoustic conditions. The "best cluster" coverage is a measure of the dispersion of a given speaker's data across clusters. We averaged the percentage of data for each speaker in the cluster which has most of his/her data. On average 80% of the speaker data is going to the same cluster. In fact, the average value is a bit misleading as there is a large variance in the best cluster coverage across speakers. For most speakers the cluster coverage is close to 100%, i.e., a single cluster covers essentially all frames of their data. However, for a few speakers (for whom there is a lot of data), the speaker is covered by two or more clusters, each containing comparable amounts of data.

In order to assess the result of automatic segmentation on the recognition performance, we ran the first decoding step (ie. no adaptation) on three evaluation data sets, using both manual (NIST) and automatic segmentations. On the eval97 and eval98 test data, the word error increase with the automatic segmentation is about 1.5% relative (0.3% absolute). A larger performance degradation was observed on the eval96 test data (2.6% relative, 0.6% absolute) due to the

long segment deleted in show 1.

ACOUSTIC MODELING

The acoustic models were trained on all the available transcribed task-specific training data, amounting to about 150 hours of audio data. This data was used to train the Gaussian mixture models needed for segmentation and the acoustic models for use in word recognition. We used the August 1997 and February 1998 releases of the LDC transcriptions. Overlapping speech portions were detected in the transcriptions and removed from the training data.

The acoustic analysis derives cepstral parameters from a Mel frequency spectrum estimated on the 0-8kHz band (0-3.5kHz for telephone speech models) every 10ms[3] For each 30ms frame the Mel scale power spectrum is computed, and the cubic root taken followed by an inverse Fourier transform. Then LPC-based cepstrum coefficients are computed. The cepstral coefficients are normalized on a segment cluster basis using cepstral mean removal and variance normalization. Each resulting cepstral coefficient for each cluster has a zero mean and unity variance. The 39-component acoustic feature vector consists of 12 cepstrum coefficents and the log energy, along with the first and second order derivatives.

Gender-dependent acoustic models were built using MAP adaptation of SI seed models for wideband and telephone band speech[2]. For computational reasons, a smaller set of acoustic models is used in the bigram pass to generate a word graph. These position-dependent, cross-word triphone models cover 5416 contexts, with 11500 tied states and 32 Gaussians per state. For trigram decoding a larger set of 27506 position-dependent, cross-word triphone models with 11500 tied states was used. Acoustic model development aimed to minimize the word error rate on the eval96 test data.

This year we used divisive decision tree clustering instead of agglomerative clustering for state-tying. This is particularly interesting when there are a very large number of states to cluster since it is at the same time both faster and is more robust than a bottom-up greedy algorithm. A set of 184 questions concern the phone position, the distinctive features (and identitites) of the phone and the neighboring phones. As was done last year, specific phone symbols are used to explicitly model filler words and breath noises.

LANGUAGE MODELING

All language models used in the different steps were obtained by interpolation of backoff n-gram language models trained on different data sets. To build the n-gram LM 4 models trained on the following sources were interpolated:

1- BN transcriptions from LDC (years 92-95) and from PSMedia (years 96 and 97 (the period 15/10/96 - 14/11/96 was excluded): 203M words

2- NAB newspaper texts and AP Wordstream texts prior to September 1995: 202M words

3- NAB newspaper texts and AP Wordstream texts from July 1996 to August 1997 (the period 15/10/96 - 14/11/96 was excluded) : 141M words

4- Transcriptions of the acoustic data, BN data (including the 1995 MarketPlace data): 1.6M words

The interpolation coefficients of these 4 LMs were chosen in order to minimize the perplexity on the Nov96 and Nov97 evaluation test sets. A backoff 4-gram LM is derived from this interpolation by merging the 4 LM components[12]. Interpolating LMs trained on the different data sets resulted in lower perplexities than training a single model on all the texts (weighted) as we have done in the past[4]. This is a better approach, both cleaner and more accurate. The perplexity of the eval97 test set with an interpolated 4-gram LM is 162.0, compared with 179.5 with a 4-gram trained on empirically weighted data. The resulting 4-gram LM is interpolated with a 3-gram class based language model, with 270 automatically determined word classes[8]. The classification procedure uses a Monte-Carlo algorithm to minimize the conditional relative entropy between a word-based bigram distribution and a class-based bigram distribution. Bigram and trigram LMs were built in a similar manner for use in the first two decoding steps.

The BN texts from PSmedia were processed using a modified version of the bn_raw2sgml.pl perl script from BBN made available by LDC. The broadcast news training texts were cleaned in order to be homogeneous with the previous texts, and filler words such as UH and UHM, were mapped to a unique form. All of the training texts (95 Hub3 and Hub4, and BN) were reprocessed to add a proportion of breath markers (4%), and of filler words (0.5%)[3]. As was done in previous years, the texts were processed so as to treat some frequent word sequences as compound words, and to treat the 1000 most frequent acronyms in the training texts as whole words instead of as sequences of independent letters.

LEXICAL MODELING

The recognition vocabulary contains 65,122 words and 72,788 phone transcriptions. All words occuring a minimum of 15 times in the broadcast news texts (63,954 words) or at least twice in the acoustic training data (23,234 words) were included in the recognition vocabulary. The lexical coverage was 99.14% and 99.53% on the eval96 and eval97 test sets respectively. The lexical coverage on a show selected from the BN texts of February 28, 1998 was 99.43%. On the eval98 test data the lexical coverage was 99.73% (99.78% on set1 / 99.67% on set2).

The lexicon was extended to include about 2000 new words (including fragments) found in the February 1998 transcription release, and 6800 words in the new training texts from PSmedia.

Pronunciations are based on a 48 phone set (3 of them are used for silence, filler words, and breath noises). The filler and breath phones were added to model these events, which are relatively frequent in the broadcast data and are not used

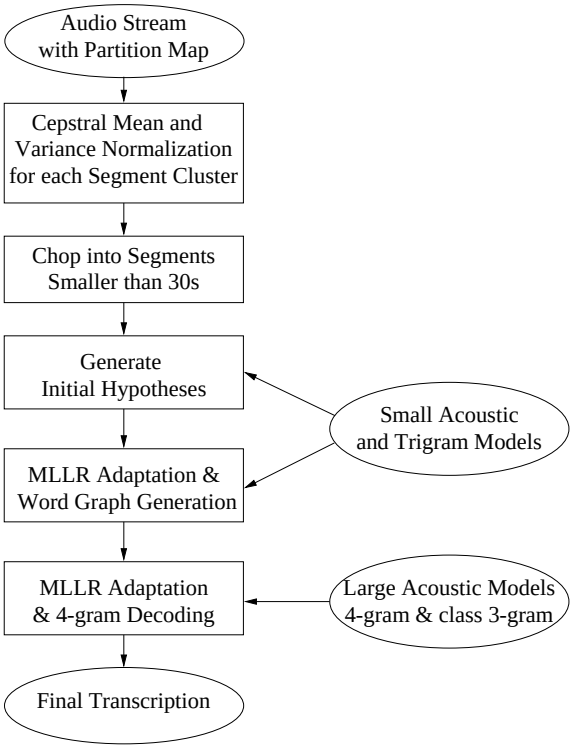

Figure 2: Word decoding.

in transcribing other lexical entries. A pronunciation graph is associated with each word so as to allow for alternate pronunciations, including optional phones. As done in previous years, the lexicon contains compound words for about 300 frequent word sequences, as well as word entries for common acronyms. This provides an easy way to allow for reduced pronunciations[3].

WORD DECODING

The word decoding procedure is shown in Figure 2. Prior to decoding, segments longer than 30s are chopped into smaller pieces so as to limit the memory required for the trigram and 4-gram decoding passes[3]. To do so a bimodal distribution is estimated by fitting a mixture of 2 Gaussians to the log-RMS power for all frames of the segment. This distribution is used to determine locations which are likely to correspond to pauses, thus being reasonable places to cut the segment. Cuts are made at the most probable pause 15s to 30s from the previous cut. Word recognition is performed in three steps: 1) initial hypothesis generation, 2) word graph generation, 3) final hypothesis generation, each with two passes.

Step 1: Initial Hypothesis Generation (fast decoding) This step, carried out in two passes, generates initial hypotheses which are used for cluster-based acoustic model adaptation. The first pass of this step generates a word graph using a small bigram backoff language model and gender-specific sets of 5416 position-dependent triphones with about 11500 tied states. This is followed by a second decoding pass with a larger set of acoustic models (27506 triphones with 11500 tied states) and a trigram language model (about 8M trigrams and 15M bigrams) to generate the hypotheses. Band-limited acoustic models are used for the telephone speech segments.

Step 2: Word Graph Generation Unsupervised acoustic model adaptation (both means and variances) is performed for each segment cluster using the MLLR technique[10]. The mean vectors are adaptated using a single block-diagonal regression matrix, and a diagonal matrix is used to adapt the variances. Each segment is decoded first with a bigram language model and an adapted version of small set of acoustic models, and then with a trigram language model (8M bigrams and 17M trigrams) and adapted versions of the larger acoustic model set.

Step 3: Final Hypothesis Generation The final hypothesis is generated using a 4-gram interpolated with a category trigram model with 270 automatically generated word classes[8]. The first pass of this step uses the large set of acoustic models adapted with the hypothesis from Step 2, and a 4-gram language model. This hypothesis is used to adapt the acoustic models prior to the final decoding step with the interpolated category trigram model.

	Test set (Word Error)		
System	*Eval96*	*Eval97*	*Eval98*
Nov96 system	**27.1***		
Nov97 system	25.3	**18.3**	
Nov98 system	19.8	13.9	**13.6**

Table 2: Summary of BN transcription word error rates. *Nov96 system used a manual partition.

Table 2 reports the word recognition results on the eval test sets from the last three years. All of our system development was carried out using the eval96 data. The results shown in bold are the official NIST scores obtained by the different systems. Only the Nov96 system used a manual partition. In Nov97 our main development effort was devoted to moving from a partitioned evaluation to the unpartitioned one. The Nov97 system did not use focus-condition specific acoustic models as had been used in the Nov96 system. This system nevertheless achieved a performance improvement of 6% on the eval96 test data. The Nov98 system has more accurate acoustic and language models, and achieves a relative word error reduction of over 20% compared to the Nov97 system.

Table 3 gives the word error rates for the Nov98 system after each decoding step and Table 4 shows the approximate computational requirements for partitioning and word decoding measured on development runs using the eval96 data. The runs were done on Silicon Graphics Origin200, R10K processor running at 180MHz and with 1Gb memory.[2] The

[2] These numbers are only indicative. No effort was made to optimize the

	Test set (Word Error)		
System Step	*Eval96*	*Eval97*	*Eval98*
Step1 3-gram	25.30	18.44	18.31
Step2 3-gram	20.95	14.56	14.24
Step3 4-gram	20.23	14.26	13.66
4-gram class	19.79	13.92	13.56

Table 3: Word error rates after each decoding step with the Nov98 system.

Step	CPU time	Memory
Partitioning:	∼2-3xRT	$<$ 10Mb
Word decoding:		
step#1 (generate tg hyp):	∼35xRT	∼300Mb
step#2 (tg run):	∼130xRT	∼400Mb
step#3 (fg + class LM):	∼30xRT	∼600Mb
Overall:	∼200xRT	

Table 4: Computational requirements on development data (eval96) with the Nov98 system.

first decoding step that is used to generate the initial hypothesis runs in about 35xRT and has a word error of 25% on the eval96 data, and 18% on the eval97 and eval98 sets. A word error reduction of about 20% is obtained in the second decoding step which uses the adapted acoustic models. Relatively small gains are obtained in the 4-gram decoding passes, even though these also include an extra acoustic model adaptation.

SUMMARY & DISCUSSION

In this paper we have presented our Nov98 broadcast news transcription system, and highlighted our development work. The main changes to our system are the generation of word graphs with adapted acoustic models using an initial hypothesis obtained in a fast decoding pass. This step is essential for obtaining word graphs with low word error rates. Unsupervised HMM adaptation is performed prior to each decoding pass using the hypothesized transcription of the previous pass. This strategy leads to a significant reduction in word error rate. The method used to generate the LMs was changed to use interpolated LMs trained on different data sets were used instead of training a single model. This led to more accurate LMs. More training data has been used for both acoustic and language modeling. Concerning the acoustic models, state-tying uses divisive decision tree clustering instead of agglomerative clustering. This is particularly interesting when there are a very large number of states to cluster. These improvements have led to a substantial performance gain (over 20%) compared to our Nov97 system. The overall word transcription error of the Nov98 unpartitioned evaluation test data (3 hours) was 13.6%. Although substantial performance improvements have been obtained, there is still plenty of room for improvement of the underlying speech recognition technology. On unrestricted broadcast news shows, such as the 1996 dev and eval data, the word error rate is still about 20% (even though the NIST scoring program has removed overlapping speech).

processing time nor the memory requirements, as long as they fit within our computational means.

*

References

[1] S.S. Chen, P.S. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion", *DARPA Broadcast News Transcription & Understanding Workshop*, pp. 127-132, Feb. 1998.

[2] J.L. Gauvain, C.H. Lee, "Maximum *a Posteriori* Estimation for Multivariate Gaussain Mixture Observation of Markov Chains," *IEEE Trans. on SAP*, **2**(2), pp. 291-298, April 1994.

[3] J.L. Gauvain, G. Adda, L. Lamel, M. Adda-Decker, "Transcribing Broadcast News: The LIMSI Nov96 Hub4 System," *ARPA Speech Recognition Workshop*, Chantilly, pp. 56-63, Feb. 1997.

[4] J.L. Gauvain, L. Lamel, G. Adda, "The LIMSI 1997 Hub-4E Transcription System", *DARPA Broadcast News Transcription & Understanding Workshop*, pp. 75-79, Feb. 1998.

[5] J.L. Gauvain, L. Lamel, G. Adda, "Partitioning and Transcription of Broadcast News Data," *ICSLP'98*, **5**, pp. 1335-1338, Sydney, Dec. 1998.

[6] H. Gish, M. Siu, R. Rohlicek, "Segregation of Speakers for Speech Recognition and Speaker Identification," *ICASSP-91*, pp. 873-876, May 1991.

[7] T. Hain, S.E. Johnson, A. Tuerk, P.C. Woodland, S.J. Young, "Segment Generation and Clustering in the HTK Broadcast News Transcription System," *DARPA Broadcast News Transcription & Understanding Workshop*, pp. 133-137, Feb. 1998.

[8] M. Jardino "Multilingual stochastic n-gram class language models," *ICASSP-96*, Atlanta, May 1996.

[9] A. Kannan, M. Ostendorf, J.R. Rohlicek, "Maximum Likelihood Clustering of Gaussians for Speech Recognition," *IEEE Trans. Speech & Audio*, **2**(3), July 1994.

[10] C.J. Leggetter, P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, **9**(2), pp. 171-185, 1995.

[11] M. Siegler, U. Jain, B. Raj, R. Stern, “Automatic Segmentation, Classification and Clustering of Broadcast News Audio,” *DARPA Speech Recognition Workshop*, Chantilly, pp. 97-99, Feb. 1997.

[12] P.C. Woodland, T. Neieler, E. Whittaker, “Language Modeling in the HTK Hub5 LVCSR”, presented at the 1998 Hub5E Workshop, Sept. 1998.