CAMBRIDGE UNIVERSITY TRANSCRIPTION SYSTEMS FOR THE MULTI-GENRE BROADCAST CHALLENGE

P.C. Woodland, X. Liu, Y. Qian, C. Zhang, M.J.F. Gales, P. Karanasou, P. Lanchantin, L. Wang

Cambridge University Engineering Dept, Trumpington St., Cambridge, CB2 1PZ U.K.
Email: {pcw,xl207,yq236,cz277,mjfg,pk407,pkl27,lw519}@eng.cam.ac.uk

ABSTRACT

We describe the development of our speech-to-text transcription systems for the 2015 Multi-Genre Broadcast (MGB) challenge. Key features of the systems are: a segmentation system based on deep neural networks (DNNs); the use of HTK 3.5 for building DNN-based hybrid and tandem acoustic models and the use of these models in a joint decoding framework; techniques for adaptation of DNN based acoustic models including parameterised activation function adaptation; alternative acoustic models built using Kaldi; and recurrent neural network language models (RNNLMs) and RNNLM adaptation. The same language models were used with both HTK and Kaldi acoustic models and various combined systems built. The final systems had the lowest error rates on the evaluation data.

Index Terms— Speech recognition, broadcast transcription, deep neural networks, HTK, Kaldi

1. INTRODUCTION

This paper describes the development of our speech-to-text transcription systems for the 2015 Multi-Genre Broadcast (MGB) challenge [4], one of the official challenge tasks at ASRU 2015. The data used was supplied by the British Broadcasting Corporation (BBC) and consists of audio from BBC television programmes. The data is very varied and covers the full range of genres (e.g. comedy, drama, sports shows, quiz shows, documentaries, news etc). The word error rate (WER) of speech-to-text (STT) systems on such data is very much higher than on the much more heavily studied broadcast news corpora [16, 49, 15] and is very variable across different genres and programme types.

Two transcription tasks were part of the challenge: standard STT in which systems process each broadcast episode as a unit and longitudinal transcription which requires systems to process multiple episodes of the same show in a causal fashion but allows information from previous episodes to be used. This paper only considers systems which use each episode as an independent unit.

The aim of the paper is to describe the development of the Cambridge MGB challenge systems. In particular we use HTK 3.5 [1, 51, 53] for building most of the models.. Aspects of particular note include an audio segmentation system based on deep neural networks (DNNs); DNN-based hybrid and tandem acoustic models and the use of these models in a joint decoding framework; various adaptation methods for DNN based acoustic models and the use of recurrent neural network language models (RNNLMs) and RNNLM adaptation. In addition we also built alternative acoustic model systems with Kaldi [33] and used the same RNNLMs. The final systems combined outputs from both HTK and Kaldi acoustic models.

The paper first gives a brief description of the data and the baseline language models used. The development of acoustic models, first on a 200 hour subset of training data is described. Based on these findings, sets of HTK models were trained on two different 700 hour training sets. The automatic segmentation system is described and evaluated, along with the acoustic models trained with Kaldi. Finally the complete systems that uses RNNLM adaptation and various combinations of acoustic models is presented.

2. DATA USED

The MGB Challenge used audio from seven weeks of television programmes with a raw total of 1600 hours of audio for acoustic model training. The audio was processed using a lightly supervised decoding process [24, 8] to extract time boundaries for utterances in the audio. Details of the data preparation for the MGB challenge are given in [4] and [25]. The output from the speech recogniser for each recognised segment was compared to the original transcript and an error rate computed between the two at either the word level (Matched Error Rate or MER) or at the phone level to yield a Phone Matched Error Rate (PMER). The maximum MER/PMER, along with an average word duration (AWD) threshold [4], was used to select data segments for training to ensure that the word/phone supervision information is reasonably accurate.

Several different training sets are used in this paper. A 200 hour set, **200h**, is initially used with HTK-based systems. This used a random selection of data with a PMER $< 20\%$. Further systems were trained with a larger corpus containing 700 hours of data, **700h-v1** with a a PMER $< 40\%$. Note all these selections also used a AWD threshold. Kaldi systems used this MGB distributed data with a maximum MER of 10% to get a 250 hour training set, **250h**, and a maximum MER of 30% to get a 500 hour set, **500h**.

After HTK systems were trained on the 700h-v1 data, the entire 1600 hours of audio was reprocessed with improved acoustic segmentation, sequence-trained hybrid acoustic models and episode based biased language models. This led to revised alignments of the BBC captions along with new MER/PMER values. In particular the amount of data with zero PMER greatly increased, and the genre balance between selections at the same PMER threshold was significantly different to the MGB distributed processing. A second selection taken from the re-processed data, with a PMER $< 30\%$, yielded a second 700 hour training set: **700h-v2**.

A large corpus of additional text data of BBC subtitles (closed-captions) was also available for the MGB challenge, yielding a total of 650 million words for language model training: 10M words of

This work is in part supported by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology). Chao Zhang is also supported by a Cambridge International Scholarship from the Cambridge Commonwealth, European & International Trust. Supporting data for this paper is available at the http://www.repository.cam.ac.uk/handle/1810/251285.

978-1-4799-7291-3/15/$31.00 ©2015 IEEE

data from the MGB 7 week acoustic transcripts; 640 million words from the additional MGB subtitle data. Text normalisation was performed to convert numeric terms into spoken forms and common abbreviations into sequences of individual letters.

A large 28 hour development test set (47 different programme episodes) for the standard transcription task **dev.full** was mainly used in this paper, as well as some results using the 12 hour longitudinal dev set (19 episodes from 5 series), **dev.long**. Some experiments use manual test set segmentations taken from the reference transcriptions.[1] The evaluation test set for Task1, **eval.std**, contained 11 hours of audio from 16 broadcast episodes. The evaluation data for Task 3, **eval.long**, contained 14 hours of audio from 19 broadcast episodes taken from only 2 series.

3. N-GRAM LANGUAGE MODELS

The two baseline 4-gram word level LMs were trained using the 650 million word text data described above. Two vocabularies were used: a 64k vocabulary covering the frequent words from the 7 week acoustic transcripts were used in initial experiments; and a larger expanded 160k vocabulary constructed by incorporating additional frequent words from both the acoustic transcripts and the subtitle LM text data. We used some manual checking and filtering of the words included. The dictionaries for both wordlists used the Combilex dictionary [34, 35] with missing pronunciations generated automatically [6].

4-gram LMs were estimated on the acoustic transcription data and subtitle data sources separately before a linear interpolation and merging was used to combine them. The interpolation weights (0.3:0.7) were perplexity optimised on the MGB transcription development set. The performance of various LMs were evaluated on the MGB dev.full data with manual segmentation. The WER calculations used hybrid acoustic models trained on the 700h-v1 set and confusion network decoding. The perplexity and WER performance of these two baseline 4-gram LMs together with the out-of-vocabulary (OOV) rates are shown as LM1 and LM2 in Table 1. For fast decoding, pruned versions these two LMs obtained by applying an entropy based pruning beam of 1.0e-9 are also shown in Table 1 as LM1$_{\rm prune}$ and LM2$_{\rm prune}$ respectively. It can be seen that the reduced OOV rate of LM2 has reduced the WER by about 0.7% absolute.

		Vocab		dev.full	
AM	LM	Size	%OOV	PPlex	%WER
700hr-v1	LM1$_{\rm prune}$	64k	1.2	108.7	25.9
hybrid	LM1			103.1	25.6
700hr-v1	LM2$_{\rm prune}$	160k	0.4	114.4	25.3
hybrid	LM2			108.6	24.9

Table 1. Perplexity, CN decoding WERs, vocabulary sizes and OOV rates of baseline 4-gram LMs on dev.full using 700hr-v1 based hybrid acoustic models and manual segmentation.

4. 200H HTK HYBRID AND TANDEM SYSTEMS

4.1. Hybrid DNN Configurations

Initially we looked at DNN input features and normalisation using the 200h training set. The standard hybrid DNN configuration used for these experiments has an output layer with 6k triphone state targets and 5 hidden layers, with each hidden layer containing 1000 units. The training uses discriminative pre-training [39] followed by "fine-tuning" with a frame based cross-entropy (CE) objective function and stochastic gradient descent (SGD).

When building DNNs for either acoustic modelling or bottleneck feature extraction, perceptual linear predictive (PLP) and log Mel-filter bank (FBK) coefficients, together with the relevant differential coefficients, are widely used as acoustic features [18, 37]. The DNN input vector at time t, $\mathbf{x}_t$, is usually formed by stacking the acoustic feature vector $\mathbf{o}_t$ with its context frame $\mathbf{o}_{t+c}$, where c is any integer from a given context shift set $\mathbf{c}$ [53]. For example, $\mathbf{c} = [-4, +4]$ will produce the input vector by concatenating $\mathbf{o}_t$ with 4 frames in its left and right contexts. In Table 2, PLP and FBK based DNN input features are compared for hybrid DNNs, with base features of 40-dimensional FBK or 13 dimensional PLP. Confusion network (CN) decoding [29, 11] is used.

Feature	$\mathbf{c}$	Mean/Variance Norm	%WER
PLP_D_A_T	$[-4, +4]$	Show-Seg/Show-Seg	33.9
FBK_D_A	$[-4, +4]$	Show-Seg/Show-Seg	31.8
FBK_D_A	$[-4, +4]$	Utterance/Show-Seg	31.6
FBK_D	$[-4, +4]$	Utterance/Show-Seg	31.5
FBK_D	$[-5, +5]$	Utterance/Show-Seg	31.6

Table 2. 200h CE DNN performance with LM1$_{\rm prune}$ on dev.full with manual segmentation. CN decoding. Differentials denoted as _D for first order, _A for second order and third order _T. Cepstral nomalisations are either per utterance or use all of the segments within a particular broadcast episode (Show-Seg).

From Table 2, all FBK systems outperformed the PLP system. Furthermore, FBK systems with only first differentials can perform as well as the FBK_D_A system. The best input configuration is FBK_D with $\mathbf{c} = [-4, +4]$, which resulted in a lower WER using fewer parameters, compared to the FBK_D_A system, and this setup was used in future experiments.

A further improvement to the speaker independent (SI) hybrid system is to use discriminative sequence training [22] which in HTK is implemented using SGD. For the initial 200h FBK_D_A hybrid system, 6 iterations of minimum phone error (MPE) training [32] were performed. The WERs of each iteration is listed in Table 3 and yields a total reduction in WER of 3.2% absolute over the CE model.

MPE Iteration	0	1	2	3	4	5
% WER	31.8	29.6	29.5	28.9	28.7	28.6

Table 3. %WER for 200h MPE sequence training hybrid DNNs on dev.full (LM1$_{\rm prune}$, manual segmentation, CN).

4.2. Tandem DNN Configurations

For tandem systems, different input configurations to the DNN for bottleneck (BN) feature extraction were tested in Table 4. Note that the input vector for the GMM-HMM acoustic models was the concatenation of i) 39-d standard acoustic features, obtained by projecting a 52-d PLP_D_A_T vector with heteroscedastic linear discriminant analysis (HLDA) to 39-d [26], and ii) the BN feature, obtained from the the DNN BN layer using a global semi-tied covariance (STC) matrix [14]. Therefore systems using BN features with FBK DNN inputs contain information from both PLP and FBK

[1] Of course, the MGB challenge required automatic audio segmentation for the evaluation.

features. We used the same decision tree for both tandem and hybrid systems and the BN-DNNs were fixed to have the same decision trees as the corresponding GMM-HMM acoustic models. As in our prior work, BN-DNNs have only one output target for silence frames [23, 28, 53].

Feature Type	Bottleneck		Hidden	%WER
/Differentials	LayerNum	Size	Activation Fn	
PLP_D_A_T	6	26	sigmoid	34.0
FBK_D_A	6	26	sigmoid	31.9
FBK_D	6	26	sigmoid	31.5
FBK_D	6	26	soft ReLU	31.1
FBK_D	5	26	soft ReLU	30.6
FBK_D	5	39	soft ReLU	30.1

Table 4. 200h SI tandem MPE performance for different DNN-BN setups. All DNN-BNs have $\mathbf{c} = [-4, +4]$. In total 6 hidden layers are used in all BN-DNNs and the all non-BN hidden layers have 1000 nodes and 6000 output units. %WER with $\text{LM1}_{\text{prune}}$ on dev.full with manual segmentation & CN decoding.

All BN-DNNs were trained with frame-based CE training, and both sigmoid and soft rectified linear unit (ReLU) hidden layer activation functions were explored for use in BN-DNNs. Table 4 compares the choice of activation function and the size and position of the bottleneck layer. The 5th hidden layer (second last) hidden layer is the best position tested and there is a further improvement from using a 39-d bottleneck feature. The systems in Table 4 all use the same lattices for MPE training generated by a non-tandem system. If the lattices are instead generated by a tandem system instead the error rate is reduced from 30.1% and 29.5% giving an overall reduction in WER for the various changes (features, activation function, bottleneck position and size, lattice generation) of 4.5% absolute. Furthermore if speaker adaptive training (SAT) [3] with constrained maximum-likelihood linear regression (CMLLR) [13] rather than SI MPE training is applied, the WER is reduced from 29.5% to 29.1%.

4.3. Joint Decoding Systems

Tandem and hybrid systems are usually highly complementary and combining such systems can result in WER reductions [44, 28, 52]. We have investigated a "joint decoding" scheme which uses a weighted combination of state-level acoustic log likelihoods from tandem and hybrid component systems. Note that a implementation restriction of our current joint decoding system is that the GMM-HMM and DNN-HMM acoustic models need to share the same decision tree [53, 28]. The symbol $\otimes$ is used to denote joint decoding.

System	%WER
Hybrid SI MPE	28.6
Tandem SI MPE	29.5
Joint: Tandem $\otimes$ Hybrid	27.4

Table 5. WER for 200h joint decoding systems on dev.full set (manual segmentation, $\text{LM1}_{\text{prune}}$, CN decoding).

Results from different system dependent combination weights were evaluated and the best joint setup listed in Table 5. It can be seen that the combination gives a 1.2% absolute reduction in WER over the sequence trained hybrid system.

5. HTK SYSTEMS USING 700H-V1 TRAINING DATA

All of the evaluation systems were based used 700 hour training data sets. Initially the 700h-v1 set was used and the procedures tuned on the 200h data set were applied to the 700h data set, but with 9.5k units in the DNN output layers. Both the SI hybrid system input and tandem BN-DNN input features are FBK_D with the context shift set$\mathbf{c} = [-4, +4]$. Thus the SI hybrid DNN acoustic model structure uses 720 (80×9) inputs, 5 hidden layers of 1000 nodes and 9.5k outputs. Only 3 iterations of DNN MPE training were completed in order to save time. The tandem DN-BNN structure again uses the same input/output sizes with a bottleneck at the 5th layer and a 39-d bottleneck, and soft ReLU hidden activation functions.

The SI MPE hybrid and tandem systems were evaluated individually and jointly decoded as shown in Table 6. Although both component systems are SI, they are still complementary due to the different activation functions, acoustic modelling methods, and extra PLP input features to the tandem system.

System	Criterion	%WER
SI Hybrid	CE	28.4
SI Hybrid	MPE	25.9
SI Tandem	MPE	27.0
Joint: Tandem $\otimes$ Hybrid	MPE	24.6

Table 6. % WER of 700h-v1 systems on dev.full ($\text{LM1}_{\text{prune}}$, manual segmentation CN decoding).

It can be seen that the WER is reduced by 2.8% absolute in joint decoding by using the 700h-v1 training set rather than the 200h set.

6. AUTOMATIC AUDIO SEGMENTATION

6.1. Speech/non-speech segmentation based on DNNs

The multi-genre broadcast data includes a wide range of acoustic conditions, and there are various sorts of non-speech included in it, such as music, applause, laughter, and various types of noise etc. For speech recognition the segmentation stage needs to partition the speech into homogeneous segments, with ideally a single speaker and audio condition so normalisation and adaptation based on segment clusters is effective.

Initial evaluation of the segmenter used in the Cambridge RT-04 broadcast news system [15], in which the first stage used GMMs of different audio types showed that it performed poorly on this data[2]. Hence an alternative architecture was investigated in which an initial speech/non-speech discrimination stage using a DNN (with a minimum duration constraint imposed with Viterbi decoding) is followed by further processing designed to ensure segment homogeneity.

A number of initial experiments were performed to determine a suitable DNN input context window size and architecture. As might be expected, it is useful to have a very wide input context window and a 55 frame window of $\mathbf{c} = [-27, +27]$ of 40-d FBK features was used, along with a first hidden layer with 1000 units and 5 further hidden layers of 200 units and a final output layer with two nodes to represent speech and non-speech. The models were all trained using the frame-based CE criterion. These were used in a Viterbi decoding framework with speech/non-speech HMMs that ensured a 2-frame minimum duration. The use of acoustic Change

[2] It was developed using US broadcast news data and would have not been allowed in an MGB challenge system.

Point Detection (CPD) which uses the likelihood of Gaussian models estimated on sliding windows to chop the initial segments further, followed by bottom-up Iterative Agglomerative Clustering (IAC) to regroup neighbouring segments was also investigated. The algorithms for CPD and IAC are taken from the Cambridge March 2005 diarisation system [41] and are also used in our MGB diarisation system [21].

A further key issue is the training data used for the DNN given that only the lightly supervised training data is available. Initially a 100 hour subset of the initial 200h training data was used and only data from the chosen speech segments used for speech and non-speech training (non-speech was the utterance internal silence only and 38h of such data was selected). This set was used to train DNN-v1 which with CPD/IAC and 50 frame utterance internal minimum silence was used in the re-processing of the BBC data to obtain the 700h-v2 set. In further models, 209 hours of data from 700h-v2 that had a zero PMER, and hence the speech/non-speech portions of training were fairly certain, was used with either 37h of utterance-internal non-speech data (37h) in DNN-v3 or in DNN-v4 using other non-aligned data to find further non speech. DNN-v4 used additional inter-segment data which was filtered by a previously trained DNN speech/non-speech system so that a total of 247h of non-speech data was used.

6.2. Segmentation performance

The performance of various segmenters with different DNNs and optionally with CPD and IAC were evaluated. WER performance is using a 700h-v1 sequence trained hybrid DNN model with LM1$_{\text{prune}}$ on the dev.full set. The missed speech (MS) and false alarm speech (FA) were also computed on the same data. The segmenters include the Cambridge RT-04 segmeter [15] and the MGB baseline segmentation (segonly) [4].

System	MaxSil	IAC+CPD	% MS	%FA	%WER
Manual	—	—	—	—	26.7
MGB-base	—	—-	3.6	3.7	30.7
RT-04	—	—	4.1	7.3	33.7
DNN-v1	50	—	2.1	5.5	30.4
DNN-v1	50	√	2.6	4.2	29.9
DNN-v3	50	√	1.7	5.4	30.0
DNN-v4	50	√	2.2	2.1	29.1
DNN-v4	40	√	2.3	2.0	28.9
DNN-v4	30	√	2.5	1.9	**28.8**
DNN-v4	20	√	2.7	1.8	28.8

Table 7. Performance of different segmentations on dev.full.

It can be seen that the best performance is given by the system based on DNN-v4 with 30 frames of internal silence in CPD and IAC stages. This gives a 1.9% reduction in WER over the MGB baseline and 4.9% absolute over the mismatched RT-04 segmenter, and was used in all subsequent experiments. Note that the same segmenter was also used for our MGB diarisation [21] and alignment [25] systems.

7. HTK SYSTEMS BASED ON 700H-V2 TRAINING DATA

As explained in Sec. 2 a revised lightly supervised alignment was performed using a preliminary DNN-based automatic segmenter and sequence-trained hybrid acoustic models trained on the 700h-v1 setup with strong episode-based biased language models [25] to yield a revised training set 700h-v2. Note that in both cases, the original BBC subtitle word sequences were used for acoustic training.[3]

The impact of the improved data processing in 700h-v2 was evaluated by training SI MPE GMM-HMMs, see Table 8. Although the reduction in WER is small, and is due to a reduction in deletion errors, all further HTK systems were trained using the 700h-v2 data.

Training Data Set	%WER
700h-v1	40.7
700h-v2	40.3

Table 8. %WER of MPE GMM-HMM (non-tandem) systems trained on 700h-v1 and 700h-v2 on dev.full (LM1$_{\text{prune}}$, manual segmentation, CN decoding).

7.1. 700h-v2 Tandem and Hybrid Systems

The input feature setup for the DNNs and tandem GMM-HMMs is the same as in Section 5. Since the 700h-v2 systems were used in the final evaluation systems, larger DNN structures were adopted. The DNN architectures for the hybrid and tandem DNNs are 720 × 2000^5 × 1000 × 12000 and 720 × 2000^4 × 39 × 2000 × 12000 respectively, and both DNNs were trained based on the alignments produced by the 3rd system in Table 6. The training lattices of the tandem SAT system were generated by the relevant SI tandem system. Only one iteration of DNN MPE training was completed since the sequence training for this large DNN is rather slow.

A 700h-v2 9.5k state joint decoding system was also constructed (SI hybrid and SI tandem). The %WER with the 160k LM2$_{\text{prune}}$ and CN decoding is 25.3% for this system with automatic segmentation.

7.2. Speaker Adaptive Stacked Hybrid System

Besides the SI hybrid system and tandem SAT system, a speaker adaptive (SA) stacked hybrid system was also built to allow alternative forms of adaptation to be used in the final system. In a similar way to other recent Cambridge systems with stacked hybrid configurations [23, 28], the SA stacked hybrid system was built directly using the 78d 700h-v2 tandem SAT features, as in Section 7.1. Furthermore, to allow the stacked hybrid system to be able to look at a longer context span, the context shift set is set to have gaps, [47, 28], i.e., $\mathbf{c} = \{-20, -15, -10, -5, 0, +5, +10, +15, +20\}$. The DNN acoustic model has a ReLU hidden layer activation function and with an input layer of 720 units, 6 hidden layers with 1000 units and 12k outputs.

Recent studies showed that adapting different types of DNN activation function parameters is an effective way to model speaker characteristics [42, 45, 55] and have been shown to be complementary to CMLLR input transforms for DNN adaptation [46]. Here, a novel DNN activation function adaptation approach based on recently proposed parameterised ReLU ($p-$ReLU) function [54] is used in the SA stacked hybrid system. A $p-\text{ReLU}(\alpha_{s,i}, \beta_{s,i})$ function is defined as

$$f_{s,i}(a) = \left\{ \begin{array}{ll} \alpha_{s,i} \cdot a & \text{if } a > 0 \\ \beta_{s,i} \cdot a & \text{if } a \leqslant 0 \end{array} \right. ,$$

[3] We were unable to investigate the use of the lightly supervised recogniser output for providing training transcripts for acoustic model training in the time available, and we leave this to future work.

where i is a hidden unit and s represents a speaker. As shown in [54], $p-$ReLU$(\alpha_{s,i}, 0)$ usually performs better than training $\beta_{s,i}$, and is used to adapt the hidden layers of the current stacked hybrid system. Similar to the procedure in [42, 45], the initial values of $\alpha_{s,i}$ for all speakers and all hidden units of the bottom layer are set to 1.0. Training $\alpha_{s,i}$ proceeds while keeping all the other DNN parameters fixed. The adaptation is performed at the sequence level but is based on the CE criterion. Gradient clipping is used to avoid potential training failures caused by gradient explosion [5]. When adapting the upper hidden layers, both the standard DNN parameters and all the $p-$ReLU function parameters of the previously adapted lower layers are kept fixed. Hence this procedure is executed in a layer-by-layer fashion. As shown in Table 9, the system with 5 hidden layers adapted has the lowest WER, and is the structure used in the final evaluation system. Note that even with CMLLR input transforms, the additional $p-$ReLU adaptation gives a further 1.1% absolute WER reduction.

Input Transform	$p-$ReLU Adaptation	%WER
CMLLR	None	25.9
CMLLR	Bottom Layer	25.5
CMLLR	Bottom 2 Layers	25.2
CMLLR	Bottom 3 Layers	25.0
CMLLR	Bottom 4 Layers	24.9
CMLLR	Bottom 5 Layers	24.8
CMLLR	Bottom 6 Layers	25.0

Table 9. %WER of 700h-v2 SA stacked hybrid system on dev.full (Automatic segmentation, 160k LM2$_{\text{prune}}$, CN decoding)

8. RNNLMS WITH TOPIC ADAPTATION

The ability of RNNLMs [30, 31], to model long span contexts has led to their increasing use in state-of-the-art LVCSR systems. In this paper, RNNLMs with a non-class based, full vocabulary output layer were efficiently trained on GPUs in a bunch mode [9]. An out-of-shortlist (OOS) node was used at the output layer to model the probability mass assigned to OOS words. All RNNLMs used a 64k word input layer vocabulary and 60k word output layer shortlist.

As RNNLMs use a vector history that extends to the start of the utterance, it is non-trivial to directly rescore ASR word lattices. Instead, N-best list rescoring is normally used [30, 40]. However in order to effectively use CN decoding we need to rescore lattices. Efficient RNNLM lattice rescoring using an n-gram style approximation of history contexts as proposed in [27] is used here. The same acoustic model and decoding setup in Table 1 is used. The perplexity and CN decoding WER performance of the baseline RNNLM with 512 hidden nodes, RNN512, are shown in 2nd line in Table 10. After equal weight based linear interpolation with the 4-gram LM1 (also previously shown in line 2 in Table 1), an absolute WER reduction of 0.6% was obtained over LM1.

Adaptive language models in which the LM parameters are altered according the current topic or data style are known to improve performance. One method to condition the RNNLM to the current topic is adding auxiliary input features, alongside the binary 1-of-k encoding of the current word to the RNNLM. Here we have applied Latent Dirichlet allocation (LDA) [7] to extract 30 dimensional episode level topic posterior vectors and fed into both the input and output layers of an RNNLM. These were used in both RNNLM training to improve their generalisation performance and facilitate an efficient topic based adaptation at test time based on a first pass decoding of the current broadcast episode [10].

AM	LM	dev.full	
		PPlex	%WER
700hr-v1 MPE hybrid	LM1	103.1	25.6
	LM1+RNN512	93.0	25.0
	LM1+RNN512.lda	85.1	24.7
	LM1+RNN1024.lda	81.0	24.4
700hr-v1 MPE hybrid	LM2	108.6	24.9
	LM2+RNN1024.lda	85.7	23.7

Table 10. Perplexity and CN decoding WER performance of baseline and topic adapted RNNLMs on dev.full using 700h-v1 MPE hybrid acoustic models, manual segmentation & CN decoding.

The performance of the 512 hidden layer node topic adapted RNNLM "RNN512.lda" is shown in the 3rd line in table 10. This topic adapted RNNLM outperformed RNN512 by 0.3% absolute. Further increasing the number of hidden layer nodes to 1024, a larger topic adapted RNN1024.lda gave an additional WER reduction of 0.3% absolute. Compared to the 160k word vocabulary baseline 4-gram LM2, the 1024 hidden node topic adapted RNN1024.lda gave an overall WER reduction of 1.2% absolute.

9. KALDI SYSTEMS

Three sets of neural network acoustic models were built using Kaldi [33]: (i) DNN; (ii) convolutional neural network (CNN); (iii) long short term memory (LSTM) hybrid systems. The overall aim was to see if alternative acoustic models with different training setups could complement those trained with HTK. For the Kaldi DNN and CNN systems, 500 hours of training data from the original MGB data was selected (MER $<$ 30%) and for LSTM, due to the longer training time only 250 hours was used (MER $<$ 10%). The distributed MGB Kaldi recipe was followed to build 39-d MFCC feature based LDA-MLLT-SAT GMM-HMM models, which were then used to generate the basic clustered state alignment for later neural network training. GMM-HMM models with 10.5k clustered context-dependent states were used for the DNN, and 9k states for the CNN and LSTM.

When developing the hybrid systems, the 40 dimensional Kaldi Mel filter bank features and 3 pitch features [17] were along with both the first and second-order derivatives. Speaker-cluster level mean and variance normalization was performed on all the feature dimensions. The final input feature for model training was formed from a context window of 11 frames creating an input layer of 1419 units ($129 * 11$) for the NN model training. The individual model configurations for these three systems are given below.

- **DNN**: 6 hidden layers with 1024 sigmoid units in each layer and the soft-max output layer has 10.5K output units.
- **CNN**: A frequency-based 1-d CNN architecture similar to that in [2], containing two convolutional hidden layers (128 feature maps in the first and 256 feature maps in the second convolutional layer, with filter sizes of 8 and 4 respectively). Non-overlapping max pooling is used. After the convolutional layers, 4 fully-connected hidden layers of 1024 nodes are arranged and the output layer size has 9k units.
- **LSTM**: This follows the strategy proposed in [36] and contains 2 LSTM hidden layers, where each LSTM layer has 512 cells, and a 200 unit projection layer for dimensionality reduction[4]. The output state label is delayed by 5 frames. The output layer has 9k targets.

[4] The size of the LSTM model is small compared to other work [36, 38] due to the long training time

All the networks were initialized using RBM pre-training [19] and “fine-tuned” using the cross-entropy criterion. The state-frame alignments were re-generated and updated using the initial CE hybrid systems, and then the models fine-tuned again. After CE training, sequence training was used to improve performance [48].

The Kaldi based systems used Minimum Bayes Risk (MBR) decoding [50], and MBR combination as shown in Table 11. All the results here used the Cambridge automatic segmentation.

System	WER	System	WER
	std lat		regen lat
CNN (K00)	26.4	CNN (K04)	26.0
DNN (K01)	27.7	DNN (K05)	27.0
LSTM (K03)	31.1	—	—
K00:K01	26.0	K04:K05	25.5
K00:K01:K03	**25.7**	K04:K05:K03	**25.3**

Table 11. WER (%) for the Kaldi CNN, DNN and LSTM individual systems and MBR combination represented by ”:” (Cambridge automatic segmentations and LM2).

The results show that the CNN hybrid system gave the lowest WER among the three single systems, and the LSTM model performs relatively poorer due to the smaller training set. MBR-based system combination reduces the error rate and it is particularly interesting that although the LSTM system WER is rather higher, it still contributes to the overall system combination performance.

The models in the left hand side of Table 11 were used to regenerate the training lattices, and then sequence training was re-run with these new lattices. The results are given in the right-hand side of Table 11[5]. Using the regenerated lattices, the individual components are improved by up to 0.7% absolute and the final combination has a 0.4% absolute reduction in WER. The best performance from the system combination achieves 25.3% and 23.4% on the MGB dev.full set and dev.long sets respectively.

10. FINAL SYSTEMS

The final systems developed used all the elements described so far. For the standard transcription evaluation, not all system components were completed by the deadline and the structure adopted was slightly different to that which was preferred. These initial systems are shown in Table 12. The system submitted as the primary submission was the best available at the deadline and was the ROVER [12] combination of the H04 system (joint 12k tandem SAT / SI hybrid 700h-v2) and the Kaldi MBR combination (i.e.H04+K00:K01:K03). Table 12 also shows the results of these systems on the eval.std evaluation data.

Further systems were developed and applied to both the dev.full and dev.long sets. This included the use of a revised initial SI pass based on 700h-v2 training (H10 in Table 13), and a joint 9.5k 700h-v2 system that uses a 30-d i-vector speaker representation (with priors [20]) in the hybrid component (H13). The Kalidi systems using regenerated lattices were used with RNNLM rescoring (systems K07, K08, K09) by converting the Kaldi lattices to HTK format and applying the topic adapted RNNLM. The resulting lattices were then converted to confusion networks and confusion network combination applied for both HTK and Kaldi based systems. Note that in this process the posterior scores in the confusion networks were mapped using system-specific piece wise linear mappings implemented via a simple decision tree.

ID	System	LM	dev.full	eval.std
H00	9.5k 700h-v1 SI	RNN512.lda	25.0	26.0
H04	Joint 12k 700-v2	RNN1024.lda	23.5	24.1
K00:K01:K03		LM2	25.7	26.7
H04+K00:K01:K03		—	23.0	**23.7**
H06	p-RELU adapt	RNN1024.lda	23.9	—
H04 ⊕ H06		RNN1024.lda	22.4	22.8

Table 12. Systems for the initial Task 1 evaluation. System H00 is used for adaptation of further systems. H04 is a joint system of tandem SAT and SI hybrid models. ⊕ represents confusion network combination; '+' ROVER combination; and ':' Kaldi MBR combination. %WER for dev.full set, and eval.std set (final scores).

ID	System	dev.full	dev.long	eval.long
H10	9.5k SI	23.9	21.7	—
H11	Joint 12k	23.3	21.1	—
H12	p-ReLU adapt	23.7	21.5	—
H11⊕H12		22.3	20.1	20.0
H13	joint 9.5k ivec	23.4	21.4	—
K07	LSTM	31.2	29.0	—
K08	CNN regen	25.5	23.9	—
K09	DNN regen	26.6	25.0	—
K08⊕K09⊕K07		25.0	22.8	21.8
H11⊕H12⊕K08		**21.8**	**19.7**	**19.4**

Table 13. Final systems giving WER (%) on dev.full and dev.long, and for a subset eval.long. All HTK-based systems trained on 700h-v2. System H10 is used for adaptation of the further systems. H11 is a joint system using tandem SAT and SI hybrid models. All systems use LM2+RNN1024.lda.

Table 13 gives various results on both dev sets and for some systems on eval.long. In addition to those systems, a six-way combination of H11⊕H12⊕H13⊕K08⊕K09⊕K07 gives **21.7**% on the dev.full set as well as **19.7**% on the dev.long set and was submitted as the primary system in the longitudinal transcription evaluation where it obtained **19.3%** WER on eval.long. The same system obtained **22.1%**WER on eval.std. Note that this system gives a 1.6% absolute lower WER on eval.std (1.3% lower on dev.full) than the H04+K00:K01:K03 system which was submitted as the Task 1 primary system.

The best HTK-only system in Table 13 is the 2-way combination of H11⊕H12 which has 0.4% to 0.6% absolute higher WER than the best six-way combination. The best system with only Kaldi acoustic models K08⊕K09⊕K07. The best three-way combination of H11⊕H12⊕K08 gives error rates very close to the Task3 Primary system (only 0.1% higher) but with considerably less complexity.

11. CONCLUSIONS

This paper has described the various techniques used to develop systems for the MGB challenge. Key features included the use of a DNN-based segmentation system; HTK-based DNN-based hybrid and tandem acoustic models in a joint decoding framework; adaptation of ReLU DNNs using parameterised activation function adaptation; recurrent neural network language models (RNNLMs) and RNNLM adaptation; and the use of alternative Kaldi models. The primary systems that we submitted for both the standard transcription and longitudinal transcription tasks had the lowest overall error rates.

[5] Note: lattice regeneration only used for DNN and CNN Kaldi models.

12. REFERENCES

[1] http://htk.eng.cam.ac.uk

[2] O. Abdel-Hamid, A. Mohamed, H. Jiang, & G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition", *Proc. ICASSP*, Kyoto, 2012.

[3] T. Anastasakos, J. McDonough, R. Schwartz, & J. Makhoul, "A compact model for speaker adaptive training", *Proc. ICSLP*, Philadelphia, 1996.

[4] P. Bell, M.J.F. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Wester & P.C. Woodland. "The MGB challenge: Evaluating multi-genre broadcast media transcription", *Proc. ASRU Workshop*, Scottsdale, 2015.

[5] Y. Bengio, P. Simard, & P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, 1994.

[6] M. Bisani & H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion, *Speech Communication*, vol. 50, no. 5, 2008.

[7] D.M. Blei, A. Ng, & M.I. Jordan, "Latent Dirichlet allocation", *Journal of Machine Learning Research*, vol. 3, pp. 99–1022, 2003.

[8] H.Y. Chan & P.C. Woodland, "Improving broadcast news transcription by lightly supervised discriminative training", *Proc. ICASSP*, Montreal, 2004.

[9] X. Chen, Y. Wang, X. Liu, M.J.F. Gales, & P.C. Woodland, "Efficient GPU-based training of recurrent neural network language models using spliced sentence bunch", *Proc. Interspeech*, Singapore, 2014.

[10] X. Chen, T. Tan, X. Liu, P. Lanchantin, M.J.F. Gales & P.C. Woodland, "Recurrent Neural Network Language Model Adaptation for Multi-Genre Broadcast Speech Recognition", *Proc. Interspeech*, Dresden, 2015.

[11] G. Evermann & P.C. Woodland, "Posterior probability decoding, confidence estimation and system combination", *Proc. Speech Transcription Workshop*, College Park, MD, 2000.

[12] J. Fiscus, "A post-processing system to yield reduced word error rates: recogniser output voting error reduction (ROVER), i*Proc. ASRU Workshop*, Santa Barbara, 1997.

[13] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition", *Computer Speech and Language*, vol. 12, pp. 75–98, 1997.

[14] M.J.F. Gales, "Semi-tied covariance matrices for hidden Markov models", *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[15] M.J.F. Gales, D.Y. Kim, P.C. Woodland, H.Y. Chan, D. Mrva, R. Sinha, & S.E. Tranter, "Progress in the CU-HTK broadcast news transcription system", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1513–1525, 2006.

[16] J.L. Gauvain, L. Lamel, & G. Adda "The LIMSI broadcast news transcription system" *Speech communication*, vol. 37, no. 1, pp. 89–108, 2002.

[17] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, & S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition", *Proc. ICASSP*, Florence, 2014.

[18] F. Grezl & P. Fousek, "Optimizing bottle-neck features for LVCSR", *Proc. ICASSP*, Las Vegas, 2008.

[19] G.E. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines", *Technical Report, UTML TR 2010-003*, Department of Computer Science, University of Toronto, 2010.

[20] P. Karanasou, M.J.F. Gales & P.C. Woodland, "I-vector estimation using informative priors for adaptation of deep neural networks", *Proc. Interspeech*, Dresden, 2015.

[21] P. Karanasou, M.J.F. Gales, P. Lanchantin, X. Liu, Y. Qian, L. Wang, P.C. Woodland & C. Zhang, "Speaker diarisation and longitudinal linking in multi-genre broadcast data", *Proc. ASRU Workshop*, Scottsdale, 2015.

[22] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling, *Proc. ICASSP*, Taipei, 2009.

[23] K.M. Knill, M.J.F. Gales, S.P. Rath, P.C. Woodland, C. Zhang, & S.-X. Zhang, "Investigation of multilingual deep neural networks for spoken term detection", *Proc. ASRU Workshop*, Olomouc, 2013.

[24] L. Lamel, J.L. Gauvain, & G. Adda, "Lightly supervised and unsupervised acoustic model training", *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[25] P. Lanchantin, M.J.F. Gales, P. Karanasou, X. Liu, Y. Qian, L. Wang, P.C. Woodland & C. Zhang, "The development of the Cambridge University alignment systems for the Multi-Genre Broadcast challenge", *Proc. ASRU Workshop*, Scottsdale, 2015.

[26] X. Liu, M.J.F. Gales, & P.C. Woodland, "Automatic complexity control for HLDA systems", *Proc. ICASSP*, Hong Kong, 2003.

[27] X. Liu, Y. Wang, X. Chen, M.J.F. Gales, & P.C. Woodland, "Efficient lattice rescoring using recurrent neural network language models", *Proc. ICASSP*, Florence, 2014.

[28] X. Liu, F. Flego, L. Wang, C. Zhang, M.J.F. Gales, & P.C. Woodland, "The Cambridge University 2014 BOLT conversational telephone Mandarin Chinese LVCSR system for speech translation", *Proc. Interspeech*, Dresden, 2015.

[29] L. Mangu, E. Brill, & A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks", *Computer Speech and Language*, Vol. 14, No. 4, pp. 373–400, 2000.

[30] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, & S. Khudanpur, "Recurrent neural network based language model", *Proc. Interspeech*, Makuhari, Japan, 2010.

[31] T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocky, & S. Khudanpur, "Extensions of recurrent neural network language model", *Proc. ICASSP*, Prague, 2011.

[32] D. Povey & P.C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training", *Proc. ICASSP*, Orlando, 2002.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M.Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovskỳ, G. Stemmer, & K. Veselỳ, "The Kaldi speech recognition toolkit", *Proc. ASRU Workshop*, Hawaii, 2011.

[34] K. Richmond, R. Clark & S. Fitt, "Robust LTS rules with the Combilex speech technology lexicon", *Proc. Interspeech*, Brighton, 2009.

[35] K. Richmond, R. Clark & S. Fitt, "On generating Combilex pronunciations via morphological analysis", *Proc. Interspeech*, Makuhari, Japan, 2010.

[36] H. Sak, A. Senior, & F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling", *Proc. Interspeech*, Singapore, 2014.

[37] T. N. Sainath, B. Kingsbury, A. Mohamed, & B. Ramabhadran, "Learning filter banks within a deep neural network framework", *Proc. ASRU Workshop*, Olomouc, 2013.

[38] T.N. Sainath, O. Vinyals, A. Senior, & Hasim Sak, "Convolutional, long short-term memory, fully connected deep neural networks", *Proc. ICASSP*, Brisbane, 2015

[39] F. Seide, G. Li, X. Chen, & D. Yu, "Feature engineering in context-dependent deep neural networks", *Proc. ASRU Workshop*, Hawaii, 2011.

[40] Y. Si, Q. Zhang, T. Li, J. Pan, & Y. Yan, "Prefix tree based n-best list re-scoring for recurrent neural network language model used in speech recognition system", *Proc. Interspeech*, Lyon, 2013.

[41] R. Sinha, S.E. Tranter, M.J.F. Gales, & P.C. Woodland, "The Cambridge University March 2005 speaker diarisation system", *Proc. Interspeech*, 2005.

[42] S.M. Siniscalchi, J.-Y. Li, & C.-H. Lee, "Hermitian polynomial for speaker adaptation of connectionist speech recognition systems", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 2152–2161, 2013.

[43] A. Stolcke, "SRILM: an extensible language modeling toolkit", *Proc. ICSLP*, Denver, 2002.

[44] P. Swietojanski, A. Ghoshal, & S. Renals, "Revisiting hybrid and GMM-HMM system combination techniques", *Proc. ICASSP*, Vancouver, 2013.

[45] P. Swietojanski & S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models", *Proc. IWSLT*, Lake Tahoe, 2014.

[46] P. Swietojanski & S. Renals, "Differentiable pooling for unsupervised speaker adaptation", *Proc. ICASSP*, Brisbane, 2015.

[47] K. Veselý, M. Karafiát, & F. Grézl, "Convolutive Bottleneck Network Features for LVCSR", *Proc. ASRU Workshop*, Hawaii, 2011.

[48] K. Vesely̌, A. Ghoshal, L. Burget, & D. Povey, "Sequence-discriminative training of deep neural networks", *Proc. Interspeech*, Lyon, 2013.

[49] P.C. Woodland, "The development of the HTK broadcast news transcription system: An overview", *Speech Communication*, vol. 37, no. 1, pp. 47–67, 2002.

[50] H. Xu, D. Povey, L. Mangu, & J. Zhu, "Minimum Bayes risk decoding and system combination based on a recursion for edit distance", *Computer Speech & Language*, vol. 25, no. 4, pp. 802–828, 2011.

[51] S.J. Young, G. Evermann, M.J.F. Gales, T. Hain., D. Kershaw, X. Liu, G. Moore, J.J. Odell, D. Ollason, D. Povey, V. Valtchev, and P.C. Woodland, *The HTK book (for HTK version 3.4)*. Cambridge University Engineering Department, 2006.

[52] D. Yu & L. Deng, "Fuse deep neural network and Gaussian mixture model systems", *Automatic Speech Recognition: A Deep Learning Approach*, pp. 177–191. Springer, London, 2015.

[53] C. Zhang & P.C. Woodland, "A general artificial neural network extension for HTK", *Proc. Interspeech*, Dresden, 2015.

[54] C. Zhang & P.C. Woodland, "Parameterised sigmoid and ReLU hidden activation functions for DNN acoustic modelling", *Proc. Interspeech*, Dresden, 2015.

[55] Y. Zhao, J.-Y. Li, J. Xue, & Y.-F. Gong, "Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data", *Proc. ICASSP*, Brisbane, 2015.