Interspeech 2018
2-6 September 2018, Hyderabad

# CRIM's System for the MGB-3 English Multi-Genre Broadcast Media Transcription

*Vishwa Gupta, Gilles Boulianne*

Centre de recherche informatique de Montréal (CRIM)
{Vishwa.Gupta, Gilles.Boulianne}@crim.ca

## Abstract

The second English Multi-Genre Broadcast Challenge (MGB-3) is a controlled evaluation of speech recognition and lightly supervised alignment using BBC TV recordings. CRIM is participating in the speech recognition part of the challenge. This paper presents CRIM's contributions to the MGB-3 transcription task. This task is inherently more difficult than the first task as the training audio has been reduced from 1200 hours to 500 hours. CRIM's main contributions are experimentation with bidirectional LSTM models and lattice-free MMI (LF-MMI) trained TDNN models for acoustic modeling, LSTM and DNN models for speech/non-speech detection for input to speaker diarization, and LSTM language models for rescoring lattices. We also show that adding senone posteriors to the input of LSTM and DNN models for speech/non-speech detection (VAD) reduces error rate. CRIM's best single decoding WER for the MGB-3 dev17 dev set (with reference segmentation) went down from 27.6% (with our MGB-1 challenge system) to 24.1% for this task using the LF-MMI trained TDNN models. The final WER on dev17 set (after VAD) is 20.9%, and on the new dev18 development set is 20.8%.

**Index Terms**: Deep Neural Networks, DNN, change point detection, automatic transcription, multi-genre broadcast transcription.

## 1. Introduction

The second English Multi-Genre Broadcast Challenge (MGB-3) is a controlled evaluation of speech recognition and *lightly supervised* alignment using BBC TV recordings [1]. CRIM is participating in the speech recognition (or transcription) part of the challenge. In MGB-3, the acoustic training data consists of 500 hours of BBC recordings of episodes from many genres. We only have closed captioned transcripts of these episodes. These transcripts have been converted into lightly supervised speech segments with word-matched error rates (WMER) or phone-matched error rates (PMER) by the MGB-3 organizers. The WMER and PMER are obtained by comparing the closed-captioned text with the output of a baseline recognizer.

In MGB-1 challenge, the best results were obtained by Cambridge University [2]. In acoustic data selection, they used a PMER threshold to select the acoustic training data for generating acoustic models. These acoustic models were then used to perform lightly supervised alignment one more time to generate another training set using a PMER threshold. They got a small reduction in word error rate (WER) through this process. In acoustic modeling, they performed joint decoding with tandem DNN acoustic models and hybrid DNN acoustic models. They also used Kaldi [3] to generate CNN and unidirectional 2-level LSTM acoustic models. In speaker segmentation, a DNN was trained to provide accurate speech/non-speech segmentation. The speech segments were then diarized and recognized using various models and then followed by model combinations.

We also used the WMER/PMER to select a training set. We train bi-directional LSTM models [4] from this training data. In the second iteration, we select a new training set as follows: we select training set by first aligning the recognized transcript with the closed-captioned text, then selecting only the aligned words that have one or more phones in common (since many of the errors correspond to similar sounding words). We also remove aligned words that have long durations to remove matches to long music segments.

In acoustic modeling, CRIM's contribution is to experiment with bidirectional LSTM [4], and LF-MMI trained TDNN models (chain models) [5]. We found that bidirectional LSTM's and the chain TDNN are significantly superior to other DNN acoustic models we tried in MGB-1 [6].

We experimented with DNNs and LSTMs for voice activity detection (VAD) in order to improve speaker diarization. We found that concatenating senone posteriors to the MFCC features input to the LSTM and DNN VADs reduce the voice activity detection errors. The best VAD results were with DNNs with a large input context. We also show that WER for segmentation with VAD is lower than that with VAD+BIC clustering.

In language modeling, besides quadgram language models, we have also tried RNNLM and LSTM LM's. Since LSTM LM's have long term memory, we have tried to decode the entire episode with LSTM, i.e., we do not reset the LSTM after every speaker turn, since the entire episode is semantically connected.

## 2. Acoustic training data selection

The acoustic training data provided by MGB challenge committee contains lightly supervised alignments based on the transcripts from closed captioning. As a measure of confidence, they also computed phone matched error rates (PMER) and word matched error rates (WMER) [1]. In the second MGB challenge for English, the total acoustic data available is 500 hours of audio, significantly less than the 1200 hours available during the first MGB challenge. The mix of genres in the dev17 and dev18 sets is similar to that of the development set for the first challenge.

There are two dev sets provided by the MGB-3 organisers: dev17 and dev18. Dev17 was provided last year and most of the results in this paper have been run on dev17. Dev18 was provided recently and we have used it more as an eval set to compare dev17 and dev18 WER.

### 2.1. CRIM's acoustic training data selection approaches

In the first MGB challenge, CRIM used the different WMER values to choose the training set [6]. In MGB-3 challenge, we have tried many different algorithms to choose the training set.



10.21437/Interspeech.2018-2079

First, we used a PMER of 40% to choose the initial training set. We generated 3-level bidirectional LSTM acoustic models (with cell dimension of 1024, hidden layer dimension of 1024, and recurrent and non-recurrent projection layer dimension of 128) from this training data. (By 3-level LSTM we mean that 3 LSTMs are stacked on top of each other, with the output of 1st LSTM going to the input of the 2nd LSTM, and the output of the 2nd LSTM going to the input of the 3rd LSTM. This is refered to as a deep structure.) We then re-aligned the training data using these models and trained another set of acoustic models. The reason is that this re-alignment and retraining leads to significant reduction in error rate. We then added more training data by using a PMER of 60% and retrained these models. The first two lines of Table 1 show WER for the dev17 set with reference segmentation provided by the organisers in the STM file.

We also experimented with many other alignments to generate a reasonable training set. With either PMER or WMER, the training set contains many spoken words that are not transcribed in the closed-caption transcripts. These transcription errors could be corrupting the models. To test this hypothesis, we experimented with training sets derived from alignment of recognized time-aligned transcripts with the corresponding closed-captioned text. The best scenario we found is to use all the aligned words with at least 1 phoneme in common. Recognized words that do not align (insertions) are removed. Also, aligned words with long durations are removed since many of them happen to be music segments. Line 3 (Table 1) shows the WER for LSTM models trained from this data (referred to as *set align*. Models generated from this *align* training set gave WER comparable to that for training set with PMER 60. To get multiple transcripts for model combination, we have used two training sets: training set with PMER60 and training set *align*.

Table 1: *%WER on dev17 set (with reference segmentation) with different methods of extracting the training set.*

| method | hours of audio | WER LSTM |
|---|---|---|
| PMER 40 | 279 | 24.9 |
| PMER 60 | 314 | 24.7 |
| aligned matching words (*set align*) | 330 | 24.8 |

# 3. Acoustic Models and Single Decoding

## 3.1. CRIM acoustic models and single decoding processes

In MGB-1 challenge, we tried two different feature parameters and many different deep neural networks (DNNs): TRAP features [7] and cepstral features transformed by an fMLLR transform per speaker. The TRAP features gave significantly lower WER than the fMLLR transformed cepstral features. TRAP features gave the best results with DNNs trained from over 747 hours of audio. We recognized the MGB-3 dev set (with reference segmentation) using the best models from MGB-1. The single decoding WER was 27.6% (see Table 2).

In order to get the best possible acoustic models, we tried two different features (40 dim MFCC and fMLLR transformed MFCC features) and two different topologies (bidirectional 3-level LSTM and TDNN chain models) for acoustic modeling. Both the bidirectional LSTM and the chain TDNN models gave the best accuracies with the MFCC features. We did not try the CNN models with jump connections as they require many layers for good accuracy resulting in slow training times [8]. The TDNN chain models have 7 hidden layers, use ReLU of size 725, and the splice indexes are: "-1,0,1 -1,0,1,2 -3,0,3 -3,0,3 -3,0,3 -6,-3,0 0". All the models have 100 dimensional i-vector input [9] [10] [11]. The i-vector extractor was computed from a subset of the training set with PMER of 40.

Table 2 shows the results with different models for decoding using a small trigram language model created from the LM data provided for MGB-3. The TDNN chain models actually gave us a small improvement over the bidirectional LSTM models, which was an unexpected result for us. The chain TDNN models were trained with LF-MMI followed by fine tuning with the word-lattice based sMBR objective function [5]. Sequence training of bidirectional LSTM models gave worse results, so we have used only CE trained LSTM models. Both the bidirectional LSTM (24.7% WER) and TDNN chain model (24.1% WER) gave lower WER than our best DNN models trained from 747 hours of audio in MGB-1 (27.6% WER). We trained LSTM models with both 1-frame MFCC input and with 5-frame MFCC input. The 5-frame input LSTM model gave lower WER.

Table 2: *WER on the dev17 set (with reference segmentation) for DNNs with i-vector input.*

| | |
|---|---|
| MGB-1 best model | 27.6% |
| 3-level LSTM 1-frame input | 25.9% |
| 3-level LSTM 5-frame input | 24.7% |
| chain TDNN | 24.1% |

# 4. Voice activity detection and speaker clustering

In the previous section, we gave all the results on dev17 development set based on reference segmentation provided in the STM file. In this section, we show results with automatic segmentation of the dev17 and dev18 development sets. We already have an in-house speaker diarization system that was trained on English broadcast news data with algorithms similar to that for the French diarization system [12]. Even though we cannot use it for official results, we tried it as a benchmark for voice activity detection (VAD). For voice activity detection with this system for dev17, we got 17.4% error rate. The high VAD error rate implies that we need to train the VAD system with the MGB-3 training data.

Cambridge University had successfully used DNN-based VAD for MGB-1 challenge [2]. To reduce VAD errors (false alarms + missed speech), we tried two different architectures for neural net based VAD: DNN architecture similar to that used in [2] with varying number of input frames, and a bidirectional LSTM with 1 to 3 levels. We also tried two different feature parameters: 40-dim MFCCs, and 40-dim MFCCs with senone posteriors added to them. The senone posteriors were generated from a bidirectional LSTM with 178 senones as outputs. To train these VAD DNN models, we generated three different training sets. In the first training set (set0), we aligned all the speech segments with zero PMER. The segments aligned to words were labeled as speech and the rest as non-speech. This resulted in 20 million speech frames and only 2 million non-speech frames. The resulting 3-level LSTM gave over 30% VAD error on dev17 due to many music and noise segments being recognized as speech. So we needed to add many more non-

speech frames for training in order to balance the speech/non-speech discrimination.

We noticed in the training data with lightly aligned supervision that intervals between speech segments with closed captioning were mostly silence or music. So we added all such segments as non-speech. Including all these frames increased the non-speech frames to 31 million frames, 1.5 times the number of speech frames. We trained DNN from two different training sets: *set1*: 20 million speech, 20 million non-speech frames, *set2*: 20 million speech, 31 million non-speech frames. *set0* is 20 million speech, 2 million non-speech frames. We trained a DNN with 55-frame MFCC features as input, 5 hidden layers, with 2000, 500, 500, 500, and 200 output nodes respectively. The softmax layer has 2 outputs (speech/non-speech). We also trained 1-level and 3-level LSTM models for speech/non-speech discrimination.

Speech/non-speech detection using the DNN/LSTM is as follows: We first label each frame as speech or non-speech based on DNN/LSTM posterior likelihoods. Consecutive speech frames are merged into one segment. Segments with less than 0.3 sec silence in between are merged. Isolated segments less than 0.2 secs are discarded. The results with various training sets and with different DNNs are shown in Table 3.

Table 3: *VAD error rates (false alarm + missed speech) for MGB-3 dev17 set with different DNN training sets and DNN architectures.*

| training set | DNN/LSTM | % VAD error |
|---|---|---|
| set0 | 3-level LSTM 5-frame input | 34.1% |
| set2 | 3-level LSTM 5-frame input | 14.7% |
| set1 | 3-level LSTM 1-frame input | 12.1% |
| set1 | 1-level LSTM 1-frame input | 12.1% |
| set1 | DNN 55-frame input | 7.1% |
| set2 | DNN 55-frame input | 7.1% |
| set2 | DNN 81-frame input | 6.6% |
| | MGB VAD | 7.1% |

We also compared MFCC features versus MFCC + senone posteriors as input features. For LSTM models, we concatenated 40-dim MFCCs with 178-dim senone posteriors. For DNNs, we concatenated 81-frames of 40-dim MFCCs with the 178 senone posteriors of the center frame. The results on dev17 are shown in Table 4.

Table 4: *VAD error rates (false alarm + missed speech) for MGB-3 dev17 set with different input features and DNN architectures trained with set2.*

| input features | DNN/LSTM | % VAD error |
|---|---|---|
| 1-frame MFCC | 1-level LSTM | 12.1% |
| 1-frame MFCC + senone posteriors | 1-level LSTM | 10.3% |
| 81-frames of MFCC | DNN | 6.6% |
| 81-frames of MFCC + center-frame senone posteriors | DNN | 6.2% |

The lowest VAD error is obtained with the DNN with 81-frames of 40-dim MFCCs and 178 senone posteriors of the center frame as input. The LSTMs probably did not do as well as DNNs because their effective memory may be less than 50 frames. We also tried using joint decision by combining likelihoods from different DNNs, but the overall gains were only small. The senone posteriors had a smaller impact with DNNs probably because we input only posteriors corresponding to the center frame.

We used the speech segments found by VAD for speaker diarization as follows: Each segment was first sub-divided into homogeneous segments by a change point detection algorithm. These segments were then revised using a Viterbi re-alignment. These Viterbi re-aligned segments were then clustered into similar segments using BIC clustering [13]. The clustered segments were then modified using Viterbi re-alignment (see [12] for details of the change point detection, Viterbi alignment and BIC clustering). These segments were then used for decoding (instead of reference segments from STM file). The segmentation after VAD only gave lower WER than segmentation after VAD + BIC clustering, even though BIC clustering reduced the diarisation error rate (DER). For example, in one case, the WER for dev17 went down from 29.3% (with BIC clustering) to 28.5% (VAD only).

For dev17 and dev18 development sets we generated 2 sets of segments: one by DNN VAD with only 81-frames of MFCCs as input (VAD-MFCC) and the other by DNN VAD with 81-frames of MFCCs and senone posteriors of the center frame as input (VAD-MFCC-SENONE). We were also provided with dev17 segmentation in the dev17 XML files provided by the organisers (referred to as MGB VAD). So all together we have 3 different segmentations for dev17 and 2 different segmentations for dev18. The comparative decoding error is shown in Table 5 for the 3 different segmentations for dev17 development set. We see that there is 11.7% absolute difference between decoding without segmentation and segmentation provided by VAD-MFCC-SENONE (compare first line with the 4th line in Table 5). The small variations in the different DNN VADs we have used cause only a small difference in WER. Basically, we gain approximately 0.2% absolute WER compared to MGB VAD. All the results are for first pass decoding with a small trigram language model. The acoustic model used is a 3-level bi-directional LSTM with 5-frame input (line 3 in Table 2).

Table 5: *WER for MGB-3 dev17 set with different VAD schemes.*

| VAD | % WER |
|---|---|
| No VAD / no diarization | 39.9% |
| MGB VAD | 28.4% |
| VAD-MFCC | 28.5% |
| VAD-MFCC-SENONE | 28.2% |

## 5. Language Models

Language models were trained on provided, normalized BBC subtitles representing 646M word tokens. The normalization is described in [14]. The hand-transcribed dev17 development set from the transcription task was used for interpolation weight tuning and perplexity evaluation. The dev17 contained 61900 word tokens after removing comments, vocal noises and post-processing acronyms. First, trigram and quadgram language models were trained on all this data, with modified Kneser-Ney smoothing, and limiting the vocabulary to the 160,000 word set provided by the MGB Challenge organisers. Their respective

perplexities were 127 and 116 on the development set (2 first lines of Table 7). The quadgram LM was slightly pruned to 24.8 M 3-grams and 57.72 M 4-grams. The trigram was more heavily pruned to 2.52 M 3-grams and 3.16 M 2-grams.

The results with quadgram rescoring followed by RNNLM rescoring are shown in Table 6 for our best chain TDNN models. The RNNLM rescoring uses N-best rescoring with $N = 200$.

Table 6: *WER for MGB-3 dev17 and dev18 sets with different VAD schemes followed by trigram decoding, quadgram (4-gram) rescoring and RNNLM rescoring.*

| VAD | trigram search | 4-gram rescore | RNNLM rescore |
|---|---|---|---|
| dev17 MGB VAD | 27.1% | 24.7% | 23.9% |
| dev17 VAD-MFCC | 27.4% | 25.0% | 24.4% |
| dev17 vad-mfcc-senone | 27.4% | 25.2% | 24.7% |
| dev18 VAD-MFCC | 27.9% | 23.8% | 23.5% |
| dev18 vad-mfcc-senone | 27.5% | 23.5% | 23.0% |

In MGB-1 challenge, we generated quadgram LM for each genre, and matched the LM with the genre of the audio file for decoding. However, [15] gives very good perplexities with large LSTM topologies, and the group with the best results in Chime4 evaluation [16] showed impressive WER reduction with LSTM LM (even though they do not give any detail about architecture, training process or use in rescoring). This prompted us to explore RNN and LSTM language models and various combinations in rescoring lattices.

In Table 7, RNN is a maximum-entropy recurrent network [17] with a 64K words vocabulary, hidden layer size 300, with 400 classes and direct connection hash size of $2 \times 10^9$. The LSTM has a vocabulary of 100K words and 2 hidden layers of size 800. Both NN LMs were trained with one half of the full training set. For LSTM training, we kept together blocks of 10 consecutive sentences to preserve across-sentence context during shuffling; in that case, increasing the number of unrolling steps in a minibatch from 20 to 40 provides an improvement (lines 5-6 of Table 7). We found that the histogram of utterance length peaks at a length of 7 words for training text (closed-captioning) and at less than 2 words for the dev17 development text (manual segmentation), a mismatch that may explain why the observed improvement is not as large as expected. We are currently investigating a modification of lattice rescoring which will not reset the LM state at the beginning of each utterance.

Table 8 show the results after rescoring lattices produced when decoding with the pruned trigram. Note that for this table, decoding was based on the manual reference segmentation rather than VAD. In each rescoring, the current LM scores in the lattice and the new LM scores are given equal weight. The best result is obtained when rescoring first with quadgram, then with RNN followed by LSTM.

## 6. Merging Decoded Lattices and CTM files

We tried different ways of merging results in order to reduce word error rate (WER). One way is to combine the lattices from different decodes and to carry out MBR decoding (or 1-best decoding) over the combined lattices. Merging lattices followed by MBR decoding always increased the WER.

Another way is to combine two lattices by lattice interpolation followed by MBR decoding or 1-best decoding. This lattice interpolation reduces WER. We interpolated lattices for

Table 7: *Train, validation and dev17 development set perplexities.*

| LM | Condition | train ppl | dev17 ppl |
|---|---|---|---|
| 3g | Kneser-Ney | 87 | 127 |
| 4g | Kneser-Ney | 61 | 116 |
| RNN | iters=3 | 57 | 108 |
| RNN | iters=6 | 50 | 99 |
| LSTM | ns=20 | 85 | 109 |
| LSTM | ns=40 | 79 | 102 |

Table 8: *WER after rescoring MGB-3 dev17 set lattices from chain TDNN models with various LMs (reference segmentation).*

| None | RNN | LSTM | 4g |
|---|---|---|---|
| 24.1% | 21.7% | 21.8% | 21.5% |

| 4g+RNN | 4g+RNN+LSTM |
|---|---|
| 20.7% | 20.6% |

each variation of the dev17 set (MGB vad, VAD-MFCC, VAD-MFCC-SENONE) with all the models. We have two chain models and two LSTM models corresponding to training sets PMER60 and *align*. For example, interpolation of the two lattices from the two chain models for *dev17 MGB vad* dataset results in one combined lattice. These combined lattices reduce the WER from 0.3% to 0.9% absolute. We have 6 such combined lattices for dev17 (3 for lattices from chain models and 3 for LSTM models). When we combine the resulting 6 ctm files using ROVER [18], we get 21.8% WER for dev17. However, combining using ROVER the 12 ctm files from individual decodes (3 data sets X 4 models) results in 21.1% WER. If we also include in ROVER the 3 ctm files from the lattice combination for the chain models (15 ctm files altogether), then we get 20.9% WER. Best chain model WER is 23.9%.

For dev18 also, the results were consistent. For dev18, we have 8 ctm files (2 datasets X 4 models). Combining with ROVER ctm files from these 8 decodes gives 20.9% WER, and adding the 2 ctm files from lattice interpolation using the chain models gives 20.8% WER. Best chain model WER is 23.0%

## 7. Conclusion

The best result we obtain is less than 21% WER over all the shows for both the dev17 and dev18 development sets. This WER is quite good considering the fact that the shows include singing, noisy talk shows, children's shows and a lot of audience noise and music. A number of improvements have lead to these results. We found that bidirectional LSTM's and chain TDNNs are significantly superior to other DNN acoustic models we tried in MGB-1, even when trained with half the data. The DNN-based voice activity detection for speech/non-speech discrimination reduced the WER from 39.9% to 28.2%. Voice activity detection (VAD) using both MFCCs and senone posteriors as input reduces VAD errors. Segmentation after VAD give lower WER than after VAD + BIC clustering. ROVER of individual ctm files from different recognizers (different models and different VADs) results in lower WER than combining lattices using lattice interpolation first and then using ROVER.

# 8. References


[1] P. Bell et. al., "The MGB challenge: Evaluating multi-genre broadcast media transcription", in Proc. ASRU 2015.

[2] P. Woodland, X. Liu, Y. Qian, C. Zhang, M. Gales, P. Karanasou, P. Lanchantin, L. Wang, "Cambridge University Transcription Systems for the Multi-Genre Broadcast Challenge", in Proc. ASRU 2015, pp. 639–646, Dec. 2015.

[3] D. Povey et. al., "The Kaldi Speech Recognition Toolkit", in Proc. ASRU 2011.

[4] A. Graves, N. Jaitly, A. Mohamed, "Hybrid Speech Recognition with Deep Bidirectional LSTM", Proc. ASRU-2013, Olomouc, Czech Republic.

[5] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, S. Khudanpur, " Purely sequence-trained neural networks for ASR based on lattice-free MMI", Proc. INTERSPEECH 2016.

[6] V. Gupta, P. Deléglise, G. Boulianne, Y. Estève, S. Meignier, A. Rousseau, "CRIM and LIUM approaches for Multi-Genre Broadcast Media Transcription", in Proc. ASRU 2015.

[7] F. Grézl, "TRAP-based Probabilistic Features for Automatic Speech Recognition", Doctoral Thesis, dept. Computer Graphics & Multimedia, Brno Univ of Technology, Brno 2007.

[8] Y. Dong, W. Xiong et al., "Deep Convolutional Neural Networks with Layer-wise Context Expansion and Attention", Proc. INTERSPEECH 2016, pp. 17–21.

[9] V. Gupta, P. Kenny, P. Ouellet, T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for French broadcast audio transcription", in Proc. ICASSP 2014, Florence, Italy.

[10] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors", in Proc. ASRU 2013, pp. 55-59.

[11] A. Senior, I. Moreno, "Improving DNN speaker independence with i-vector inputs", in Proc. ICASSP 2014.

[12] V. Gupta, G. Boulianne, P. Kenny, P. Ouellet, and P. Dumouchel, "Speaker Diarization of French Broadcast News", in Proc. ICASSP 2008, pp. 4365–4368.

[13] S. Chen, P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion,", in Proc. DARPA Broadcast News Transcription and Understanding Workshop, Landsdowne, VA, USA, February 1998.

[14] P.J. Bell, F. McInnes, S. Gangireddy, M. Sinclair, A. Birch, and S. Renals, "The UEDIN english ASR system for the IWSLT 2013 evaluation", in Proc. International Workshop on Spoken Language Translation, 2013.

[15] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," in arXiv:1602.02410v2, 2016.

[16] J. Du, Y. Tu, et al., "The USTC-iFlytek System for CHiME-4 Challenge", in Proc. CHiME-4 workshop.

[17] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Cernocky, Strategies for training large scale neural network language models, in Proc. ASRU 2011, pp. 196201.

[18] J. Fiscus, "A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," In Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, 1997, pages 347354, Santa Barbara, CA, USA.