ON THE IMPORTANCE OF EVENT DETECTION FOR ASR

David Haws, Dimitrios Dimitriadis, George Saon, Samuel Thomas, Michael Picheny

IBM T.J. Watson Research Center, Yorktown Heights, USA

{dhaws,dbdimitr,gsaon,sthomas,picheny}@us.ibm.com

ABSTRACT

The performance of modern large vocabulary continuous speech recognition (LVCSR) systems is heavily affected by segment boundaries, proper speaker identification of the segments, as well as removal of spurious data. We propose to use Long Short Term Memory (LSTM) recurrent neural networks to partition audio into speech segments as well as track speaker turns. Additionally, we train an LSTM to also identify music segments. We show that the accurate detection of events, along with removal of silence and music, using our LSTM yields a 9-10% relative improvement in ASR performance. Secondary processing by speaker clustering provides an additional boost in accuracy. Event detection accuracy of the LSTM approach is also described.

Index Terms— Event Detection, Diarization, Automatic Speech Recognition, Long Short Term Memory, Music Detection.

1. INTRODUCTION

State-of-the-art LVCSR systems convert audio files into corresponding text taking under consideration different error sources, such as speech spontaneity, speaker variability, training/testing mismatch, limited resources of data for language modeling, etc. An unsupervised speaker adapted module is considered a standard component [1] for some of these systems, providing performance improvements for repeating speakers within a single session. The motivation behind this is to allow further improvements in recognition accuracy, by further improving the speakers statistics and transforms [2]. This type of meta-information, i.e. attributing audio segments to particular speakers and incrementally refining the speaker dependent transforms, can be provided by the speaker diarization process [3].

The first step in diarization is *event detection* or *segmentation* as it is widely known, where transitions from silence to speech or music, speaker turns, and speech to music or silence are detected [4]. Then, speaker clustering is performed, where all speech segments are assigned to specific speaker labels. Speaker clustering methods can be divided into online and offline categories based on processing requirements. For the case of online speaker clustering or speaker tracking, the speech segments are assigned to speaker labels as soon as the next change point is detected. It is suitable mainly for real-time transcription systems as it allows speech segments with corresponding speaker labels to be used for speech recognition with very low latency. The output from online speaker diarization can then be used for incremental speaker adaptation [5].

The baseline system for segmentation consists of two sequential actions: first, a speech activity detector (SAD) is used to find the regions of speech in the audio stream. The quality of the subsequent tasks depends greatly on the precision of this first step, since the errors will propagate. Therefore, an accurate SAD is crucial for the successful deployment of any system. The system proposed here detects and classifies the non-speech segments into different acoustic classes such as silence, music and speech. The second step in baseline systems is to perform speaker clustering to find homogeneous segments.

ASR Input Structure	SI	SA
Automatic SAD	27.12	24.69
No spkr/gender, music in stats and decode	23.69	25.82
No spkr/gender, music in stats	23.69	25.12
No spkr/gender, no music	22.05	24.23
Gender info only, no music	22.05	20.14
Speaker info, no music in stats or decode	21.09	18.02

Table 1. A summary of word error rate for ASR performance under varying levels of oracle information on internal IBM data which contained multiple speakers, automated messages, and music.

The most popular criterion for speaker segmentation is the Bayesian information criterion (BIC) [6, 7] derived from the Generalized Likelihood Ratio (GLR) [8]. The BIC criterion is used to decide whether two models represent different data samples adequately, while penalizing more complex models. This algorithm searches for change points within a window using a penalized likelihood ratio test of whether the data in the window is better modeled by a single distribution (no change point) or two different distributions (change point). The dissimilarity between the two neighboring analysis windows is estimated and thresholded. In BIC-based speaker segmentation, the choice of the thresholds and the analysis window lengths are quite important, making its performance very sensitive to these parameters. This type of sensitivity along with the increased computational complexity often makes the BIC algorithm intractable.

Herein, we propose using an LSTM-based system first to segment the audio into speech segments and in parallel, track the speaker turns. The LSTM recurrent neural network provides an internal memory of the previous states, tracking changes easily. Additionally we train an LSTM to also identify music segments and evaluate its performance of ASR.

Accurate event detection coupled with speaker clustering can dramatically improve ASR. As motivation, we performed a series of ASR experiments on internal IBM telephony conversational data where the level of oracle information was varied. (See Table 1). It is clear that removing music, adding gender information, and finally knowing full speaker information provides a clear advantage providing 22.2% and 30.2% relative improvement in Word Error Rate (WER) for a Speaker Independent (SI) system a Speaker Adapted (SA) system respectively.

The remainder of the paper is organized as follows: Section 2 describes the NNs and LSTMs used to train event detection models.

Section 3 describes the speaker independent and speaker adapted ASR systems. Section 4 describes how the LSTM and NN models were trained for event detection. Subsection 4.1 contains results on event detection accuracy on the 2000 hour Switchboard (SWB) holdout set. Subsection 4.2 then details a variation of our LSTM model where music is added to the training data and a new 'music' label is added to the LSTM output targets. ASR accuracy using a baseline SAD, BIC, are other semi-oracle approaches are compared to the segmentation results of LSTM. Finally we conclude with a discussion in Section 5.

2. NEURAL NETWORK FOR EVENT DETECTION

Deep Neural Networks (DNN) and Recurrent Neural Networks (RNN) have been used successfully in speech recognition [9, 10, 11, 12, 13, 14]. Deep neural networks possess many well-known advantages such as modeling flexibility and ability to capture low-level characteristics in lower layers while modeling non-linearities at higher levels. RNNs also are able to model complex non-linearities, and they contain recurrent (cyclic) connections which make them very adept at modeling sequence data compared to typical feed-forward DNNs. Comparatively, DNNs can handle a limited amount of temporal information via a fixed sliding window, whereas RNNs have internal states which dynamically allows previous time-steps to influence the current time step.

Long Short Term Memory (LSTM) neural networks were developed to overcome some modeling weaknesses in RNNs [15]. Additionally, LSTM do not suffer from the vanishing gradient problem which hinders RNNs [16]. LSTMs contain *memory blocks* which itself contains memory cells with self-connections. The memory cells store temporal information of the network and the network also contains special gate units to control input or output of information. Specifically the input gate controls input activation into the cell and the output gate controls the output activation of the cell. The forget gate [17] was added to LSTMs to address a problem with processing continuous data not segmented into subsequences. The forget gate allows for forgetting or resetting the memory cell's internal state, based on the input state, cell state, and previous output state. Lastly, peephole connections were added to LSTMs which allow controlled timing of output of the memory block [14]. For recent work on LSTMs and acoustic modeling see [14, 18]. See Figure 1 for a layout of the LSTM memory block.

We propose to use LSTMs in order to find events in audio streams, such as changes from speaker to silence, speaker to speaker, and silence to speaker. Additionally we train LSTM models to also find music. The design of LSTMs allows for its internal memory structure to potentially learn some previous context and use its internal cell states to assist in deciding the label of a current frame. This is opposed to a simple NN which would have to decide at a current frame (with some context) what label to predict.

3. SPEAKER INDEPENDENT VS. SPEAKER ADAPTED ASR SYSTEMS

The ASR systems used for our experiments are similar to those described in [19]. The acoustic modeling process starts with training of traditional HMM-GMM based acoustic models. The GMM models are trained on 13 dimensional PLP features estimated in 25 ms windows of speech. The cepstral features from 9 consecutive frames are then spliced after speaker based cepstral mean-variance and vocal tract length normalizations (VTLN). An LDA transform

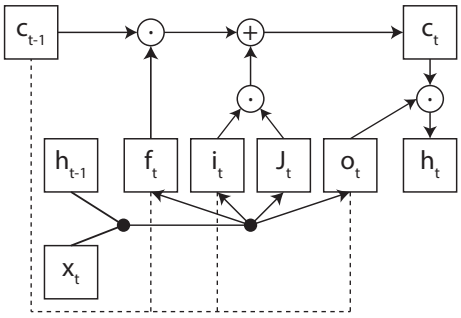

Fig. 1. A Long Short Term Memory block. The symbols x_t, c_t, h_t are the values of the input, cell, and output at time t. The symbols f_t, i_t, j_t, o_t are the forget gate, input gate, cell input activation function, and the output gate. The cells new value is determined by the addition of two values: the element-wise product $\odot$ of the previous cell state and the forget gate, and $i_t \odot j_t$. The output is then determined by the element-wise product of the cell state and the output gate. Lastly, the dashed lines show the cell peephole connections.

is applied to reduce the final feature dimensionality to 40. The ML training of the GMM models is also interleaved with the estimation of a global semi-tied covariance (STC) transform. Feature space maximum likelihood regression (FMLLR) is finally applied to train speaker adapted models. The training is done on close to 2000 hours of telephone speech from various sources including the Switchboard, Fisher and CallHome corpora.

The speaker adapted (SA) system in our experiments uses a DNN and a convolutional neural network (CNN) acoustic model combined at the score level. The DNN models are fully connected multilayer perceptrons with several non-linear hidden layers that are discriminatively trained to estimate posterior probabilities of context-dependent states. Using the standard error back-propagation and cross-entropy objective function, the DNNs are trained on speaker adapted FMLLR features using alignments produced from the HMM-GMM acoustic model described earlier. Extra speaker information is integrated by using I-Vector features as well. These features are generated as described in [20]. The DNNs are pre-trained by growing them layer-wise to 7 hidden layers. Except for the penultimate bottleneck (BN) layer with 512 units all the other hidden layers have 2048 units.

The CNN use additional feature extracting layers based on 2-D convolution before a DNN. We train CNN models on 40 dimensional VTLN warped log-mel spectra augmented with Δ and $\Delta\Delta$s. Each frame of speech is also appended with a context of ± 5 frames. All of the 128 nodes in the first feature extracting layer are attached with 9×9 filters while the second feature extracting layer with 256 nodes has a similar set of 4×3 filters. The non-linear outputs from the second feature extracting layer are then passed onto the following DNN layers. Both the DNN and CNN predict scores for 32K context dependent states, which are combined before being used for decoding. Both the DNN and CNN models are also retrained with Hessian-free sequence training as described in [21].

Unlike the speaker adapted system described above, the speaker independent (SI) system uses only a single CNN based acoustic model. The model uses an architecture similar to the SA-CNN system described earlier. Instead of speaker compensated log-mel spectra, the model uses log-mel features with utterance level normalization. This model is also retrained with Hessian-free sequence training after cross-entropy training.

The off-the-shelf SAD used for our experiments was developed for the DARPA RATS SAD program [22]. At the core of the detector is an MLP trained on a combination of several acoustic features. Speech and non-speech regions are determined by applying a Viterbi decode on scores from the model as described in [22].

4. EXPERIMENTS

The experiments were performed on the 300 hour subset of the Switchboard English conversational telephone speech (SWB) data and tested on 2.4 hours of hand transcribed internal IBM telephony data, containing multiple speakers per session, music, beeps/rings, and automated messages. The SWB data was used for training, and another 2000 hour holdout set was used to measure the speaker turn accuracy. The IBM internal data was used for testing of the ASR accuracy. Two approaches were investigated using the NNs to output homogeneous segments.

First, features, i.e. PLPs with their time-derivatives and LDA, were extracted from the training set, i.e. the SWB 300 hour data. For each utterance, features were extracted and their mean and variance were computed. Then the set of means and variances were clustered using the k-Means and the Mahalanobis distance measure into either 12, 21, or 42 clusters. That is, each SWB utterance is assigned to a cluster based on the aforementioned features. The assumption is that every cluster roughly represents a group of speakers. Various NN and LSTM architectures were, then, trained on these features, using the assigned labels per frame, i.e. the k-means cluster ID for its corresponding utterance, or silence (determined by the existing alignment).

The second and simpler approach is based on training the NN or LSTM on SWB 300 hours, where the label at each frame was either: 0 if the frame corresponded to the first speaker in the sentence, 1 if the frame corresponded to the second speaker in the sentence, or 2 for silence (determined by an existing alignment). Although the assignment of speaker 1 and speaker 2 is arbitrary, this was intentional, enabling the LSTM to rely on its dynamic ability to store information over time to determine speaker turns.

An ergodic HMM was used taking as input the DNN or LSTM posteriors, in order to smoothen the temporal trajectories. Each state (speaker cluster or speaker 0/1, and silence) corresponds to a chain of five nodes on the finite-state machine, thus forcing at least five consecutive similar states.

4.1. Event Detection & Classification

The event detection accuracy was measured on the held-out SWB 2000 hour set. Each SWB sentence was scanned left-to-right for speaker segments of length > 1.0 secs. In the case where another segment from a different speaker within 0.25 secs was found, it was considered as an *event*. The hypothesis given by the NN or LSTM were scanned for a change of segment types within 0.125 sec of the event, and marked as a *true positive* if found. False positives are ambiguous to measure and thus the number of segments-per-second were reported as a close analogy, which is also an important measure with respect to ASR since segment granularity can harm accuracy.

Surprisingly, an LSTM with a single layer and only 32 hidden nodes performed the best yielding nearly 94% accuracy while producing approximately 0.67 segments per second. A 3-layer 1024 hidden node NN trained on the simple $\{0, 1, 2\}$ labels performed nearly as well as the LSTM. The baseline system, based on BIC [6, 7], was also tested and found to perform similarly to the NN. On the contrary, a 3-layer 1024 hidden node NN trained on 21 cluster IDs performed the worst achieving 75% accuracy at the cost of nearly three times as many segments per second. [1]

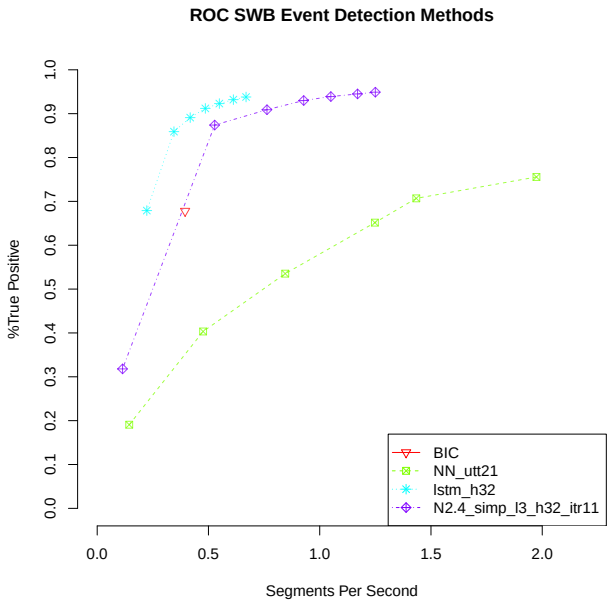

Fig. 2. ROC plots showing accuracy of event detection on SWB 2000hour data using various NN and LSTM event detection models.

4.2. ASR Accuracy & Music

Sixteen hours of music were added to the 300 hour SWB training data, a fourth label was used to specify music, and LSTMs were trained as before. The LSTMs where then used to find segments as well as remove music and silence on internal IBM telephony data consisting of 2.43 hours of conversational audio. Table 2 shows the classification of the LSTM hypothesis (first three rows) as well as how the reference (truth) were assigned by the LSTM. Interestingly the LSTM sometimes confuses music or silence and vice-versa. However, this does not present a problem as those are the two segments to be removed before ASR. The LSTM finds the majority of music, although it does mistake some speech for music and misses some music.

In Table 3 ASR performance was tested on IBM internal data using the output from the LSTM speech/silence/music segmenter, the LSTM segmenter output coupled with speaker clustering, as well as other insightful semi-oracle tests. Speaker clustering was performed using the methods described in [23]. Using the results of the LSTM directly leads to a 12.3% and 9% relative improvement in WER for the SI and SA systems respectively. Speaker clustering can improve this slightly in the SA system (SI system only estimates stats per utterance, and thus cannot benefit from speaker labeling). As a comparison, a simple SAD, even coupled with the speaker clustering does not perform as well as the LSTM (even without speaker clustering). Similar results hold for a simple SAD combined with the ubiquitous BIC. To assess the source of the difference of the WER for LSTM compared to the manual (best) two oracle tests

[1] Many NN and LSTM structures were also trained and tested. NN structures were either 1 or 3 layers with 32 or 1024 hidden units per layer. LSTMs were 1 or 3 layers with either 32, 64, 128, or 512 hidden units.

	Music	Silence	Speech
Hyp Music	49.8%	23.3%	26.9%
Hyp Silence	2.7%	92.7%	4.5%
Hyp Music&Silence	22.8%	63.0%	14.1%
Ref Music	70.4%	5.1%	24.5%
Ref Silence	15.2%	80.7%	4.1%
Ref Music&Silence	32.8%	56.6%	10.6%

Table 2. Classification accuracy of hypothesis Music, Silence, Speech labels as well as the distribution of the reference(truth) into hypothesis labels. E.g. the LSTM Music hypothesis covered 49.8% actual music, 23.3% actual silence, and 26.9% actual speech. Conversely, reference(true) music was assigned by the LSTM to be 70.4% music, 5.1% silence, and 24.5% speech.

ASR Input Structure	SI	SA
SAD	26.60	25.90
SAD + Spk. Clust. (2 spk)	26.60	24.28
SAD + Spk. Clust. (3 spk)	26.60	24.05
SAD + BIC	26.82	26.14
SAD + BIC + Spk. Clust. (2 spk)	26.82	24.05
SAD + BIC + Spk. Clust. (3 spk)	26.82	24.19
SAD (no music decode)	25.60	25.62
LSTM	23.78	23.49
LSTM + Spk. Cluster.	23.78	23.27
LSTM‡(music removed)	22.56	23.44
LSTM‡+ Spk. Clust. (3 spk)	22.56	22.47
LSTM‡+ Spk. Clust. (2 spk)	22.56	20.02
Oracle Segm. + Spk. Clust. (2 spk)	21.34	19.81
Oracle Segm. + Spk. Clust. (3 spk)	21.34	18.94
Manual (best)	21.34	18.02

Table 3. A summary of word error rate for ASR performance on internal IBM data. Data was processed with standard energy based SAD (SAD), BIC, manually in various configurations, or through the LSTM segmenter. †All stats conversation based. ‡LSTM with remaining music manually removed.

were performed. First, the remaining music was removed from the LSTM output and WER was evaluated leading to a $1-3\%$ improvement. This implies the remaining music still poses some problems for ASR. To assess the effect of the segment size and boundaries, the ground truth segments were used and speaker clustering was performed. These two tests showed that the LSTM based segmentation causes a 1.2% degradation and the speaker clustering module introduces approximately 1% degradation.

5. DISCUSSION

In this paper we demonstrated that the accuracy of ASR systems can be improved by: 1) accurate event detection and speaker segmentation, 2) removal of spurious data such as music. Additionally, passing accurate speaker segments through an online or offline speaker clustering procedure further improves ASR accuracy. To accomplish speaker segmentation we showed that an LSTM can be trained to accurately find events in continuous speech and can be adapted to also remove the majority of music present in audio streams. For future work we propose to improve the LSTM models accuracy in detecting music through changes in the network structure and the use of alternate features.

6. REFERENCES

[1] P. Cerva, J. Silovsky, J. Zdansky, J. Nouza, and L. Seps, "Speaker-adaptive speech recognition using speaker diarization for improved transcription of large spoken archives," *Speech Communication*, vol. 55, no. 10, pp. 1033–1046, 2013.

[2] J.-L. Gauvain, L. Lamel, and G. Adda, "Partitioning and transcription of broadcast news data," in *ICSLP'98*. ISCA, 1998.

[3] M. Moattar and M. Homayounpour, "A review on speaker diarization systems and approaches," 2012, vol. 54, pp. 1065–1103, Elsevier.

[4] C. Barras, X. Zhu, S. Meignier, and J.L. Gauvain, "Multistage speaker diarization of broadcast news," 2006, vol. 14, pp. 1505–1512, IEEE.

[5] D. Liu, D. Kiecza, A. Srivastava, and F. Kubala, "Online speaker adaptation and tracking for real-time speech recognition," in *Interspeech'05*. ISCA, 2005.

[6] B. Zhou and J.H.L. Hansen, "Efficient audio stream segmentation via the combined T2 statistic and the Bayesian information criterion," *Trans. on Speech and Audio Process.*, , no. 4, pp. 467–474, 2005.

[7] S.S. Chen and P.S. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," in *ICASSP'98*. IEEE, 1998.

[8] S. Sian Cheng and H. Wang S., "A sequential metric-based audio segmentation method via the Bayesian information criterion," in *Eurospeech'03*. ESCA, 2003.

[9] G. E Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[10] G. Hinton, L. Deng, D. Yu, G. E Dahl, A. Mohamed, Na. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[11] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[12] F. Eyben, M. Wöllmer, Bj. Schuller, and A. Graves, "From speech to letters-using a novel neural network architecture for grapheme based ASR," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 376–380.

[13] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[14] F. A Gers, N. N Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *The Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2003.

[15] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] F. A Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[18] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2014.

[19] H. Soltau, G. Saon, and T.N. Sainath, "Joint training of convolutional and non-convolutional neural networks," in *IEEE ICASSP*, 2014.

[20] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using I-vectors," in *IEEE ASRU*, 2013.

[21] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of neural network acoustic models using distributed Hessian-free optimization," in *ISCA Interspeech*, 2012.

[22] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system for the DARPA RATS program," in *ISCA Interspeech*, 2013.

[23] Weizhong Zhu and Jason Pelecanos, "Online speaker diarization using adapted I-vector transforms," in *ICASSP 2016*, Shanghai, China, 2016.