THE MICROSOFT 2017 CONVERSATIONAL SPEECH RECOGNITION SYSTEM

W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, A. Stolcke

Microsoft AI and Research

Technical Report MSR-TR-2017-39
August 2017

ABSTRACT

We describe the 2017 version of Microsoft's conversational speech recognition system, in which we update our 2016 system with recent developments in neural-network-based acoustic and language modeling to further advance the state of the art on the Switchboard speech recognition task. The system adds a CNN-BLSTM acoustic model to the set of model architectures we combined previously, and includes character-based and dialog session aware LSTM language models in rescoring. For system combination we adopt a two-stage approach, whereby subsets of acoustic models are first combined at the senone/frame level, followed by a word-level voting via confusion networks. We also added a confusion network rescoring step after system combination. The resulting system yields a 5.1% word error rate on the 2000 Switchboard evaluation set.

1. INTRODUCTION

We have witnessed steady progress in the improvement of automatic speech recognition (ASR) systems for conversational speech, a genre that was once considered among the hardest in the speech recognition community due to its unconstrained nature and intrinsic variability [1]. The combination of deep networks and efficient training methods with older neural modeling concepts [2, 3, 4, 5, 6, 7, 8] have produced steady advances in both acoustic modeling [9, 10, 11, 12, 13, 14, 15] and language modeling [16, 17, 18, 19]. These systems typically use deep convolutional neural network (CNN) architectures in acoustic modeling, and multi-layered recurrent networks with gated memory (long-short-term memory, LSTM [8]) models for both acoustic and language modeling, driving the word error rate on the benchmark Switchboard corpus [20] down from its mid-2000s plateau of around 15% to well below 10%. We can attribute this progress to the neural models' ability to learn regularities over a wide acoustic context in both time and frequency dimensions, and, in the case of language models, to condition on unlimited histories and learn representations of functional word similarity [21, 22].

Given these developments, we carried out an experiment last year, to measure the accuracy of a state-of-the-art conversational speech recognition system against that of professional transcribers. We were trying to answer the question whether machines had effectively caught up with humans in this, originally very challenging, speech recognition task. To measure human error on this task, we submitted the Switchboard evaluation data to our standard conversational speech transcription vendor pipeline (who was left blind to the experiment), postprocessed the output to remove text normalization discrepancies, and then applied the NIST scoring protocol. The resulting human word error was 5.9%, not statistically different from the 5.8% error rate achieved by our ASR system [23]. In a follow-up study [24], we found that qualitatively, too, the human and machine transcriptions were remarkably similar: the same short function words account for most of the errors, the same speakers tend to be easy or hard to transcribe, and it is difficult for human subjects to tell whether an errorful transcript was produced by a human or ASR. Meanwhile, another research group carried out their own measurement of human transcription error [25], while multiple groups reported further improvements in ASR performance [25, 26]. The IBM/Appen human transcription study employed a more involved transcription process with more listening passes, a pool of transcribers, and access to the conversational context of each utterance, yielding a human error rate of 5.1%. Together with a prior study by LDC [27], we can conclude that human performance, unsurprisingly, falls within a range depending on the level of effort expended.

In this paper we describe a new iteration in the development of our system, pushing well past the 5.9% benchmark we measured previously. The overall gain comes from a combination of smaller improvements in all components of the recognition system. We added an additional acoustic model architecture, a CNN-BLSTM, to our system. Language modeling was improved with an additional utterance-level LSTM based on characters instead of words, as well as a dialog session-based LSTM that uses the entire preceding conversation as history. Our system combination approach was refined by combining predictions from multiple acoustic models at both the senone/frame and word levels. Finally, we added an LM rescoring step after confusion network creation, bringing us to an overall error rate of 5.1%, thus surpassing the human

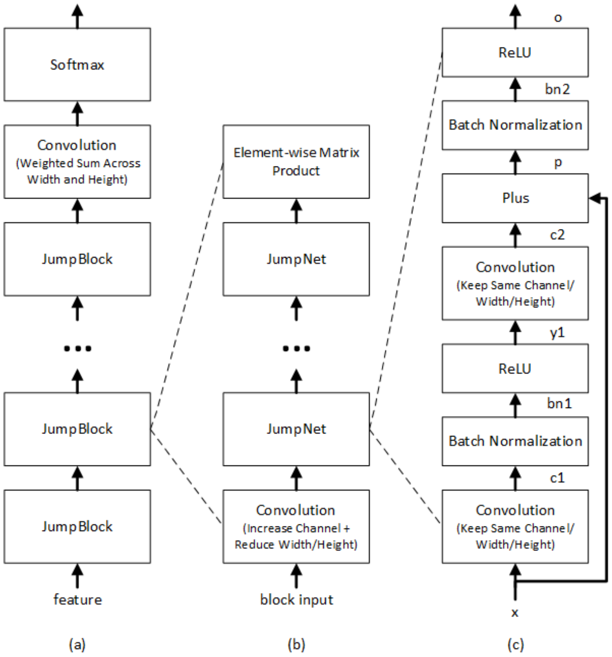

Fig. 1. LACE network architecture

accuracy level we had measured previously. The remainder of the paper describes each of these enhancements in turn, followed by overall results.

2. ACOUSTIC MODELS

2.1. Convolutional neural nets

We used two types of CNN model architectures: ResNet and LACE (VGG, a third architecture used in our previous system, was dropped). The residual-network (ResNet) architecture [28] is a standard CNN with added highway connections [29], i.e., a linear transform of each layer's input to the layer's output [29, 30]. We apply batch normalization [31] before computing rectified linear unit (ReLU) activations.

The LACE (layer-wise context expansion with attention) model is a modified CNN architecture [32]. LACE, first proposed in [32] and depicted in Figure 1, is a variant of time-delay neural network (TDNN) [4] in which each higher layer is a weighted sum of nonlinear transformations of a window of lower layer frames. Lower layers focus on extracting simple local patterns while higher layers extract complex patterns that cover broader contexts. Since not all frames in a window carry the same importance, a learned attention mask is applied, shown as the "element-wise matrix product" in Figure 1. The LACE model thus differs from the earlier TDNN models [4, 33] in this attention masking, as well as the ResNet-like linear pass-through connections. As shown in the diagram, the model is composed of four blocks, each with the same architecture. Each block starts with a convolution layer with stride two, which sub-samples the input and increases the number of channels. This layer is followed by four ReLU convolution layers with jump-links similar to those used in ResNet. As for ResNet, batch normalization [31] is used between layers.

2.2. Bidirectional LSTM

For our LSTM-based acoustic models we use a bidirectional architecture (BLSTM) [34] without frame-skipping [11]. The core model structure is the LSTM defined in [10]. We found that using networks with more than six layers did not improve the word error rate on the development set, and chose 512 hidden units, per direction, per layer; this gave a reasonable trade-off between training time and final model accuracy.

BLSTM performance was significantly enhanced using a spatial smoothing technique, first described in [23]. Briefly, a two-dimensional topology is imposed on each layer, and activation patterns in which neighboring units are correlated are rewarded.

2.3. CNN-BLSTM

A new addition to our system this year is a CNN-BLSTM model inspired by [35]. Unlike the original BLSTM model, we included the context of each time point as an input feature in the model. The context windows was $[-3, 3]$, so the input feature has size 40x7xt, with zero-padding in the frequency dimension, but not in the time dimension. We first apply three convolutional layers on the features at time t, and then apply six BLSTM layers to the resulting time sequence, similar to structure of our pure BLSTM model.

Table 1 compares the layer structure and parameters of the two pure CNN architectures, as well as the CNN-BLSTM.

2.4. Senone set diversity

One standard element of state-of-the-art ASR systems is the combination of multiple acoustic models. Assuming these models are *diverse*, i.e., make errors that are not perfectly correlated, an averaging or voting combination of these models should reduce error. In the past we have relied mainly on different model architectures to produce diverse acoustic models. However, results in [23] for multiple BLSTM models showed that diversity can also be achieved using different sets of senones (clustered subphonetic units). Therefore, we have now adopted a variety of senone sets for all model architectures. Senone sets differ by clustering detail (9k versus 27k senones), as well as two slightly different phone sets and corresponding dictionaries. The standard version is based on the CMU dictionary and phone set (without stress, but including a schwa phone). An alternate dictionary adds specialized vowel and nasal phones used exclusively for filled pauses and backchannel words, inspired by [36]. Combined with set sizes, this gives us a total of four distinct senone sets.

Table 1. Comparison of CNN layer structures and parameters

	ResNet	LACE	CNN-BLSTM
Number of parameters	38M	65M	48M
Number of weight layers	49	22	10
Input	40x41	40x61	40x7xt
Convolution_1	[conv 1x1, 64 conv 3x3, 64 conv 1x1, 256] x 3	jump block [conv 3x3, 128] x 5	[conv 3x3, 32, padding in feature dim.] x 3
Convolution_2	[conv 1x1, 128 conv 3x3, 128 conv 1x1, 512] x 4	jump block [conv 3x3, 256] x 5	
Convolution_3	[conv 1x1, 256 conv 3x3, 256 conv 1x1, 1024] x 6	jump block [conv 3x3, 512] x 5	
Convolution_4	[conv 1x1, 512 conv 3x3, 512 conv 1x1, 2048] x 3	jump block [conv 3x3, 1024] x 5	
BLSTM			[blstm, cells = 512] x 6
Output	average pool Softmax (9k or 27k)	[conv 3x4, 1] x 1 Softmax (9k or 27k)	Softmax (9k or 27k)

2.5. Speaker adaptation

Speaker adaptive modeling in our system is based on conditioning the network on an i-vector [37] characterization of each speaker [38, 39]. A 100-dimensional i-vector is generated for each conversation side (channel A or B of the audio file, i.e., all the speech coming from the same speaker). For the BLSTM systems, the conversation-side i-vector v_s is appended to each frame of input. For convolutional networks, this approach is inappropriate because we do not expect to see spatially contiguous patterns in the input. Instead, for the CNNs, we add a learnable weight matrix W^l to each layer, and add $W^l v_s$ to the activation of the layer before the nonlinearity. Thus, in the CNN, the i-vector essentially serves as an speaker-dependent bias to each layer. For results showing the effectiveness of i-vector adaptation on our models, see [40].

2.6. Sequence training

All our models are sequence-trained using maximum mutual information (MMI) as the discriminative objective function. Based on the approaches of [41] and [42], the denominator graph is a full trigram LM over phones and senones. The forward-backward computations are cast as matrix operations, and can therefore be carried out efficiently on GPUs without requiring a lattice approximation of the search space. For details of our implementation and empirical evaluation relative to cross-entropy trained models, see [40].

2.7. Frame-level model combination

In our new system we added frame-level combination of senone posteriors from multiple acoustic models. Such a combination of neural acoustic models is effectively just another, albeit more complex, neural model. Frame-level model combination is constrained by the fact that the underlying senone sets must be identical.

Table 2 shows the error rates achieved by various senone set, model architectures, and frame-level combination of multiple architectures. The results are based on N-gram language models, and all combinations are equal-weighted.

3. LANGUAGE MODELS

3.1. Vocabulary size

In the past we had used a relatively small vocabulary of 30,500 words drawn only from in-domain (Switchboard and Fisher corpus) training data. While this yields an out-of-vocabulary (OOV) rate well below 1%, our error rates have reached levels where even small absolute reductions in OOVs could potentially have a significant impact on overall accuracy. We supplemented the in-domain vocabulary with the most frequent words in the out-of-domain sources also used for language model training: the LDC Broadcast News corpus and the UW Conversational Web corpus. Boosting the vocabulary size to 165k reduced the OOV rate (excluding word fragments) on the eval2002 devset from 0.29% to 0.06%. Devset error rate (using the 9k-senones BLSTM+ResNet+LACE acoustic models, see Table 2) dropped from 9.90% to 9.78%.

Table 2. Acoustic model performance by senone set, model architecture, and for various frame-level combinations, using an N-gram LM. The "puhpum" senone sets use an alternate dictionary with special phones for filled pauses.

Senone set	Architecture	devset WER	test WER
9k	BLSTM	11.5	8.3
	ResNet	10.0	8.2
	LACE	11.2	8.1
	CNN-BLSTM	11.3	8.4
	BLSTM+ResNet+LACE	9.8	7.2
	BLSTM+ResNet+LACE+CNN-BLSTM	9.6	7.2
9k puhpum	BLSTM	11.3	8.1
	ResNet	11.2	8.4
	LACE	11.1	8.3
	CNN-BLSTM	11.6	8.4
	BLSTM+ResNet+LACE	9.7	7.4
	BLSTM+ResNet+LACE+CNN-BLSTM	9.7	7.3
27k	BLSTM	11.4	8.0
	ResNet	11.5	8.8
	LACE	11.3	8.8
	BLSTM+ResNet+LACE	10.0	7.5
27k puhpum	BLSTM	11.3	8.0
	ResNet	11.2	8.0
	LACE	11.0	8.4
	BLSTM+ResNet+LACE	9.8	7.3

3.2. LSTM-LM rescoring

For each acoustic model our system decodes with a slightly pruned 4-gram LM and generates lattices. These are then rescored with the full 4-gram LM to generate 500-best lists. The N-best lists in turn are then rescored with LSTM-LMs.

Following promising results by other researchers [43, 19], we had already adopted LSTM-LMs in our previous system, with a few enhancements [23]:

- Interpolation of models based on one-hot word encodings (with embedding layer) and another model using letter-trigram word encoding (without extra embedding layer).
- Log-linear combination of forward- and backward-running models.
- Pretraining on the large out-of-domain UW Web corpus (without learning rate adjustment), followed by final training on in-domain data only, with learning rate adjustment schedule.
- Improved convergence through a variation of self-stabilization [44], in which each output vector x of nonlinearities are scaled by $\frac{1}{4}\ln(1+e^{4\beta})$, where a β is a scalar that is learned for each output. This has a similar effect as the scale of the well-known batch normalization technique [31], but can be used in recurrent loops.
- Data-driven learning of the penalty to assign to words that occur in the decoder LM but not in the LSTM-LM vocabulary. The latter consists of all words occurring twice or more in the in-domain data (38k words).

Also, for word-encoded LSTM-LMs, we use the approach from [45] to tie the input embedding and output embedding together.

In our updated system, we add the following additional utterance-scoped LSTM-LM variants:

- A character-based LSTM-LM
- A letter-trigram word-based LSTM-LM using a variant version of text normalization
- A letter-trigram word-based LSTM-LM using a subset of the full in-domain training corpus (a result of holding out a portion of training data for perplexity tuning)

All LSTM-LMs with word-level input use three 1000-dimensional hidden layers. The word embedding layer for the word-based is also of size 1000, and the letter-trigram encoding has size 7190 (the number of unique trigrams). The character-level LSTM-LM uses two 1000-dimensional hidden layers, on top of a 300-dimensional embedding layer.

As before, we build forward and backward running versions of these models, and combine them additively in the log-probability space, using equal weights. Unlike before, we combine the different LSTM-architectures via log-linear combination in the rescoring stage, rather than via linear interpolation at the word level. The new approach is more convenient when the relative weighting of a large number of models

Table 3. Perplexities of utterance-scoped LSTM-LMs

Model structure	Direction	PPL devset	PPL test
Word input, one-hot	forward	50.95	44.69
	backward	51.08	44.72
Word input, letter-trigram	forward	50.76	44.55
	backward	50.99	44.76
+ alternate text norm	forward	52.08	43.87
	backward	52.02	44.23
+ alternate training set	forward	50.93	43.96
	backward	50.72	44.36
Character input	forward	51.66	44.24
	backward	51.92	45.00

needs to be optimized, and the optimization happens jointly with the other knowledge sources, such as the acoustic and pronunciation model scores.

Table 3 shows perplexities of the various LSTM language models on dev and test sets. The forward and backward versions have very similar perplexities, justifying tying their weights in the eventual score weighting. There are differences between the various input encodings, but they are small, on the order of 2-4% relative.

3.3. Dialog session-based modeling

The task at hand is not just to recognize isolated utterances but entire conversations. We are already exploiting global conversation-level consistency via speaker adaptation, by extracting i-vector from all the speech on one side of the conversation, as described earlier. It stands to reason that the language model could also benefit from information beyond the current utterance, in two ways: first conversations exhibit global coherence, especially in terms of conversation topic, and especially since Switchboard conversations are nominally on a pre-defined topic. Lexical entrainment [46], the tendency of conversants to adopt the same words and phrases, could also be exploited for language modeling. There is a large body of work on adaptation of language models to topics and otherwise [47], and recurrent neural network model extensions have been proposed to condition on global context [17]. A second set of conversation-level phenomena operate beyond the utterance, but more locally. Linguistic conversation analysis has long noted that utterance types come in *adjacency pairs* [48], with preferences for certain pairs over others (like a statement followed by agreement versus disagreement). Therefore, words in an utterance should be more predicable based on the previous utterance, as well as information about whether a speaker change occurred, and this has been proposed as useful information in language models [49]. We can also include in the history information on whether utterances overlap, since overlap is partially predicted by the words spoken [50]. This type of conditioning information could help model dialog behaviors such as floor grabbing, back-channeling, and collaborative completions.

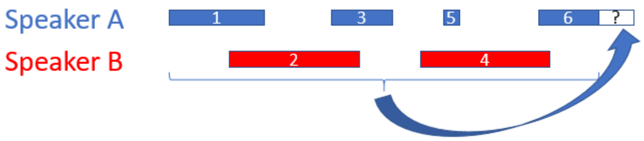

Fig. 2. Use of conversation-level context in session-based LM

Table 4. Perplexities and word errors with session-based LSTM-LMs (forward direction only). The last line reflects the use of 1-best recognition output for words in preceding utterances.

Model inputs	PPL devset	PPL test	WER devset	WER test
Utterance words, letter-3grams	50.76	44.55	9.5	6.8
+ session history words	39.69	36.95		
+ speaker change	38.20	35.48		
+ speaker overlap	37.86	35.02		
(with 1-best history)	40.60	37.90	9.3	6.7

In order to capture both global and local context for modeling the current utterance, we train session-based LSTM-LMs. We serialize the utterances in a conversation based on their onset times (using the waveform cut points as approximate utterance onset and end times). We then string the words from both speakers together to predict the following word at each position, as depicted in Figure 2. Optionally, extra bits in the input are used to encode whether a speaker change occurred, or whether the current utterance overlaps in time with the previous one. When evaluating the session-based LMs on speech test data, we use the 1-best hypotheses from the N-best generation step (which uses only an N-gram LM) as a stand-in for the conversation history.

Table 4 shows the effect of session-level modeling and of these optional elements on the model perplexity. There is a large perplexity reduction of 21% by conditioning on the previous word context, with smaller incremental reductions from adding speaker change and overlap information. The table also compares the word error rate with the full session-based model to the baseline, within-utterance LSTM-LM. As shown in the last row of the table, some of the perplexity gain over the baseline is negated by the use of 1-best recognition output for the conversation history. However, the perplexity degrades by only 7-8% relative due to the noisy history.

For inclusion in the overall system, we built letter-trigram and word-based versions of the session-based LSTM (in both directions). Both session-based LM scores are added to the utterance-based LSTM-LMs described earlier for log-linear combination.

4. EXPERIMENTAL SETUP

4.1. Data

The data sets used for system training are unchanged [23]; they consist of the public and shared data sets used in the DARPA research community. Acoustic training used the English CTS (Switchboard and Fisher) corpora. Language model training, in addition, used the English CallHome transcripts, the BBN Switchboard-2 transcripts, the LDC Hub4 (Broadcast News) corpus, and the UW conversational web corpus [51]. Evaluation is carried out on the NIST 2000 CTS test set Switchboard portion. The Switchboard-1 and Switchboard-2 portions of the NIST 2002 CTS test set were used for tuning and development.

4.2. Model training

All neural networks in the final system were trained with the Microsoft Cognitive Toolkit, or CNTK [52, 53] on a Linux-based multi-GPU server farm. CNTK allows for flexible model definition, while at the same time scaling very efficiently across multiple GPUs *and* multiple servers. The resulting fast experimental turnaround using the full 2000-hour corpus was critical for our work.

Training the acoustic models in this paper on a single GPU would take many weeks or even months. CNTK made training times feasible by parallelizing the stochastic gradient descent (SGD) training with a *1-bit SGD* parallelization technique [54]. This data-parallel method distributes minibatches over multiple worker nodes, and then aggregates the sub-gradients. While the necessary communication time would otherwise be prohibitive, the 1-bit SGD method eliminates the bottleneck by combining 1-bit quantization of gradients and automatic minibatch-size scaling, as described in more detail in [23].

We use the CNTK "FsAdaGrad" learning algorithm, which is an implementation of Adam [55]. A typical learning rate is 3×10^{-6}, and learning rates are automatically adjusted with a decrease factor of 0.7. Momentum is set at a constant value of 2500 throughout model training. For individual acoustic models, we find that training converges after 1.5 to 2 passes over the 2000-hour training set. We do not use dropout or gradient noise in our model training, only the aforementioned spatial smoothing technique for BLSTM model training.

5. SYSTEM COMBINATION AND RESULTS

5.1. Confusion network combination

After rescoring all system outputs with all language models, we combine all scores log-linearly and normalize to estimate utterance-level posterior probabilities. All N-best outputs for the same utterance are then concatenated and merged into a single word confusion network (CN), using the SRILM nbest-rover tool [56, 36].

5.2. System Selection

Unlike in our previous system, we do not apply estimated, system-level weights to the posterior probabilities estimated from the N-best hypotheses. All systems have equal weight upon combination. This simplification allows us to perform a brute-force search over all possible subsets of systems, picking the ones that give the lowest word error on the development set. We started with 9 of our best individual systems, and eliminated two, leaving a combination of 7 systems,

5.3. Confusion network rescoring and backchannel modeling

As a final processing step, we generate new N-best lists from the confusion networks resulting from system combination. Following [57], these are once more rescored using the N-gram LM, but also with a subset of the utterance-level LSTM-LMs, and one additional knowledge source. The word log posteriors from the confusion network take the place of the acoustic model scores in this final rescoring step.

The additional knowledge source at this stage was motivated by our analysis of differences between machine versus human transcription errors [24]. We found that the major machine-specific error pattern is a misrecognition of filled pauses ('uh', 'um') as backchannel acknowledgments ('uh-huh', 'mhm'). In order to allow the system learn a correction for this problem, we provide the number of backchannel tokens in a hypotheses as a pseudo-score and allow the score weight optimization to find a penalty for it. (Indeed, a negative weight is learned for the backchannel count.)

Table 5 compares the individual systems that were selected for combination, before and after rescoring with LSTM-LMs, and then shows the progression of results in the final processing stages, starting with the LM-rescored individual systems, the system combination, and the CN rescoring. The collection of LSTM-LMs (which includes the session-based LMs) gives a very consistent 22 to 25% relative error reduction on individual systems, compared to the N-gram LM. The system combination reduces error by 4% relative over the best individual systems, and the CN rescoring improves another 2-3% relative.

6. CONCLUSIONS AND FUTURE WORK

We have described the latest iteration of our conversational speech recognition system. The acoustic model was enhanced by adding a CNN-BLSTM system, and the more systematic use of a variety of senone sets, to benefit later system combination. We also switched to combining different

Table 5. Results for LSTM-LM rescoring on systems selected for combination, the combined system, and confusion network rescoring

Senone set	Model/combination step	WER devset	WER test	WER devset	WER test
		ngram-LM		LSTM-LMs	
9k	BLSTM	11.5	8.3	9.2	6.3
27k	BLSTM	11.4	8.0	9.3	6.3
27k-puhpum	BLSTM	11.3	8.0	9.2	6.3
9k	BLSTM+ResNet+LACE+CNN-BLSTM	9.6	7.2	7.7	5.4
9k-puhpum	BLSTM+ResNet+LACE	9.7	7.4	7.8	5.4
9k-puhpum	BLSTM+ResNet+LACE+CNN-BLSTM	9.7	7.3	7.8	5.5
27k	BLSTM+ResNet+LACE	10.0	7.5	8.0	5.8
-	Confusion network combination			7.4	5.2
-	+ LSTM rescoring			7.3	5.2
-	+ ngram rescoring			7.2	5.2
-	+ backchannel penalty			7.2	5.1

model architectures first at the senone/frame level, resulting in several acoustic combined systems that are then fed into the confusion-network-based combination at the word level. The language model was updated with larger vocabulary (lowering the OOV rate by about 0.2% absolute), additional LSTM-LM variants for rescoring, and most importantly, session-level LSTM-LM that can model global and local coherence between utterances, as well as dialog phenomena. The session-level model gives over 20% relative perplexity reduction. Finally, we introduce a confusion network rescoring step with special treatment for backchannels (based on a prior error analysis), that gives a small additional gain after systems are combined. Overall, we have reduced error rate for the Switchboard tasks by 12% relative, from 5.8% for the 2016 system, to now 5.1%. We note that this level of error is on par with the multi-transcriber error rate previously reported on the same task.

Future work we plan from here includes a more thorough evaluation, including on the CallHome genre of speech. We also want to gain a better understanding of the linguistic phenomena captured by the session-level language model, and reexamine the differences between human transcriber and machine errors.

Acknowledgments. We wish to thank our colleagues Hakan Erdogan, Xiaodong He, Jinyu Li, Frank Seide, Mike Seltzer, and Takuya Yoshioka for their valued input during system development, and ICSI for assistance with CTS data sets.

7. REFERENCES

[1] S. Greenberg, J. Hollenback, and D. Ellis, "Insights into spoken language gleaned from phonetic transcription of the Switchboard corpus", *in Proc. ICSLP*, 1996.

[2] F. J. Pineda, "Generalization of back-propagation to recurrent neural networks", *Physical Review Letters*, vol. 59, pp. 2229, 1987.

[3] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks", *Neural Computation*, vol. 1, pp. 270–280, 1989.

[4] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks", *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, pp. 328–339, 1989.

[5] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series", *The handbook of brain theory and neural networks*, vol. 3361, pp. 1995, 1995.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", *Neural computation*, vol. 1, pp. 541–551, 1989.

[7] T. Robinson and F. Fallside, "A recurrent error propagation network speech recognition system", *Computer Speech & Language*, vol. 5, pp. 259–274, 1991.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[9] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks", *in Proc. Interspeech*, pp. 437–440, 2011.

[10] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling", *in Proc. Interspeech*, pp. 338–342, 2014.

[11] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition", *in Proc. Interspeech*, pp. 1468–1472, 2015.

[12] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The IBM 2015 English conversational telephone speech recognition system", *in Proc. Interspeech*, pp. 3140–3144, 2015.

[13] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR", *in Proc. IEEE ICASSP*, pp. 4955–4959. IEEE, 2016.

[14] M. Bi, Y. Qian, and K. Yu, "Very deep convolutional neural networks for LVCSR", *in Proc. Interspeech*, pp. 3259–3263, 2015.

[15] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition", *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 24, pp. 2263–2276, Aug. 2016.

[16] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model", *in Proc. Interspeech*, pp. 1045–1048, 2010.

[17] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model", *in Proc. Interspeech*, pp. 901–904, 2012.

[18] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling", *in Proc. Interspeech*, pp. 194–197, 2012.

[19] I. Medennikov, A. Prudnikov, and A. Zatvornitskiy, "Improving English conversational telephone speech recognition", *in Proc. Interspeech*, pp. 2–6, 2016.

[20] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development", *in Proc. IEEE ICASSP*, vol. 1, pp. 517–520. IEEE, 1992.

[21] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models", *in Studies in Fuzziness and Soft Computing*, vol. 194, pp. 137–186. 2006.

[22] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations", *in HLT-NAACL*, vol. 13, pp. 746–751, 2013.

[23] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition", Technical Report MSR-TR-2016-71, Microsoft Research, Oct. 2016, https://arxiv.org/abs/1610.05256.

[24] A. Stolcke and J. Droppo, "Comparing human and machine errors in conversational speech transcription", *in Proc. Interspeech*, pp. 137–141, Stockholm, Aug. 2017.

[25] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines", *in Proc. Interspeech*, pp. 132–136, Stockholm, Aug. 2017.

[26] K. J. Han, S. Hahm, B.-H. Kim, J. Kim, and I. Lane, "Deep learning-based telephony speech recognition in the wild", *in Proc. Interspeech*, pp. 1323–1327, Stockholm, Aug. 2017.

[27] M. L. Glenn, S. Strassel, H. Lee, K. Maeda, R. Zakhary, and X. Li, "Transcription methods for consistency, volume and efficiency", *in Proc. 7th Intl. Conf. on Language Resources and Evaluation*, pp. 2915–2920, Valleta, Malta, 2010.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", arXiv preprint arXiv:1512.03385, 2015.

[29] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks", *CoRR*, vol. abs/1505.00387, 2015.

[30] P. Ghahremani, J. Droppo, and M. L. Seltzer, "Linearly augmented deep neural network", *in Proc. IEEE ICASSP*, pp. 5085–5089. IEEE, 2016.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *Proceedings of Machine Learning Research*, vol. 37, pp. 448–456, 2015.

[32] D. Yu, W. Xiong, J. Droppo, A. Stolcke, G. Ye, J. Li, and G. Zweig, "Deep convolutional neural networks with layer-wise context expansion and attention", *in Proc. Interspeech*, pp. 17–21, 2016.

[33] A. Waibel, H. Sawai, and K. Shikano, "Consonant recognition by modular construction of large phonemic time-delay neural networks", *in Proc. IEEE ICASSP*, pp. 112–115. IEEE, 1989.

[34] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures", *Neural Networks*, vol. 18, pp. 602–610, 2005.

[35] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks", *in Proc. IEEE ICASSP*, 2015.

[36] A. Stolcke et al., "The SRI March 2000 Hub-5 conversational speech transcription system", *in Proceedings NIST Speech Transcription Workshop*, College Park, MD, May 2000.

[37] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification", *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[38] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors", *in IEEE Speech Recognition and Understanding Workshop*, pp. 55–59, 2013.

[39] G. Saon, T. Sercu, S. J. Rennie, and H. J. Kuo, "The IBM 2016 English conversational telephone speech recognition system", *in Proc. Interspeech*, pp. 7–11, Sep. 2016.

[40] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The Microsoft 2016 conversational speech recognition system", *in Proc. IEEE ICASSP*, pp. 5255–5259, 2017.

[41] S. F. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program", *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, pp. 1596–1608, 2006.

[42] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI", *in Proc. Interspeech*, pp. 2751–2755, 2016.

[43] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 517–529, Mar. 2015.

[44] P. Ghahremani and J. Droppo, "Self-stabilized deep neural network", *in Proc. IEEE ICASSP*, pp. 5450–5454. IEEE, 2016.

[45] O. Press and L. Wolf, "Using the output embedding to improve language models", arXiv preprint arXiv:1608.05859, 2016.

[46] S. E. Brennan and H. H. Clark, "Conceptual pacts and lexical choice in conversation", *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 22, pp. 1482–1493, 1996.

[47] J. R. Bellegarda, "Statistical language model adaptation: review and perspectives", *Speech Communication*, vol. 42, pp. 93–108, 2004.

[48] E. A. Schegloff, "Sequencing in conversational openings", *American Anthropologist*, vol. 70, pp. 1075–1095, 1968.

[49] G. Ji and J. Bilmes, "Multi-speaker language modeling", in S. Dumais, D. Marcu, and S. Roukos, editors, *Proceedings of HLT-NAACL 2004, Conference of the North American Chapter of the Association of Computational Linguistics*, vol. Short Papers, pp. 133–136, Boston, May 2004. Association for Computational Linguistics.

[50] E. Shriberg, A. Stolcke, and D. Baron, "Observations on overlap: Findings and implications for automatic processing of multi-party conversation", in P. Dalsgaard, B. Lindberg, H. Benner, and Z. Tan, editors, *Proceedings of the 7th European Conference on Speech Communication and Technology*, vol. 2, pp. 1359–1362, Aalborg, Denmark, Sep. 2001.

[51] I. Bulyko, M. Ostendorf, and A. Stolcke, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures", in M. Hearst and M. Ostendorf, editors, *Proceedings of HLT-NAACL 2003, Conference of the North American Chapter of the Association of Computational Linguistics*, vol. 2, pp. 7–9, Edmonton, Alberta, Canada, Mar. 2003. Association for Computational Linguistics.

[52] D. Yu et al., "An introduction to computational networks and the Computational Network Toolkit", Technical Report MSR-TR-2014-112, Microsoft Research, 2014, https://github.com/Microsoft/CNTK.

[53] Microsoft Research, "The Microsoft Cognition Toolkit (CNTK)", https://cntk.ai.

[54] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs", *in Proc. Interspeech*, pp. 1058–1062, 2014.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", Proceedings 3rd International Conference for Learning Representations, arXiv preprint arXiv:1703.02136, May 2015.

[56] A. Stolcke, "SRILM—an extensible language modeling toolkit", *in Proc. Interspeech*, pp. 2002–2005, 2002.

[57] S. Bangalore, G. Bordel, and G. Riccardi, "Computing consensus translation from multiple machine translation systems", *in Proceedings IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 351–354, Madonna di Campiglio, Italy, Dec. 2001.