Neural Speech-to-Text Language Models for Rescoring Hypotheses of DNN-HMM Hybrid Automatic Speech Recognition Systems

Tomohiro Tanaka, Ryo Masumura, Takafumi Moriya and Yushi Aono
NTT Media Intelligence Laboratories, NTT Corporation, Japan
E-mail: tomohiro.tanaka.ht@hco.ntt.co.jp

***Abstract*—In this paper, we propose to leverage end-to-end automatic speech recognition (ASR) systems for assisting deep neural network-hidden Markov model (DNN-HMM) hybrid ASR systems. The DNN-HMM hybrid ASR system, which is composed of an acoustic model, a language model and a pronunciation model, is known to be the most practical architecture in ASR field. On the other hand, much attention has been paied in recent studies to the end-to-end ASR systems that are fully composed of neural networks. It is known that they can yield comparative performance without introducing heuristic operations. However, one problem is that the end-to-end ASR systems sometimes suffer from redundant generation and ommission of important words in text generation phases. This is because these systems cannot explicitly consider the connection between the input speech and the output text. Therefore, our idea is to regard the end-to-end ASR systems as neural speech-to-text language models (NS2TLMs) and to use them for rescoring hypotheses generated in the DNN-HMM hybrid ASR systems. This enables us to leverage the end-to-end ASR systems while avoiding the generation issues because the DNN-HMM hybrid ASR systems can generate speech-aligned hypotheses. It is expected that the NS2TLMs improve the DNN-HMM hybrid ASR systems because the end-to-end ASR systems correctly handle short-duration utterances. In our experiments, we use state-of-the-art DNN-HMM hybrid ASR systems with convolutional and long short-term memory recurrent neural network acoustic models and end-to-end ASR systems based on attetional encoder-decoder. We demonstrate that our proposed method can yield a better ASR performance than both the DNN-HMM hybrid ASR system and the end-to-end ASR system.**

I. Introduction

In modern automatic speech recognition (ASR) technologies, deep neural network-hidden Markov model (DNN-HMM) hybrid ASR systems are known to be the most practical implementation [1]. The DNN-HMM hybrid ASR systems are composed of an acoustic model, a language model and a pronunciation model, each of which is individually represented as an isolated architecture. Various studies have examined the possibility of enhancing the acoustic models and the language models. In the acoustic modeling, various DNN topologies including convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been studied for accurately converting the input speech into a phonetic sequence [2]–[5]. In language modeling, neural language models have been developed in order to improve the traditional n-gram models [6], [7]. Among the neural language models, recurrent neural network based language models (RNNLMs) [8], [9] have been largely studied as they can capture variable length word contexts, which help to improve ASR performance.

On the other hand, much attention has been paid to the end-to-end ASR systems that are fully composed of neural networks [10]–[14]. These systems have shown competitive ASR performance compared to the DNN-HMM hybrid ASR systems. In fact, the end-to-end ASR systems simultaneously learn the acoustic and the language models from the acoustic data and its transcriptions. They model generative probabilities of words or characters conditioned on the acoustic features extracted from speech. Therefore, end-to-end ASR systems can directly generate texts from speech without introducing heuristic operations.

However, end-to-end ASR systems sometimes suffer from redundant generation and omission of important words in text generation phases. This is because they cannot explicitly consider the connection between the input speech and the output text. In fact, these generation issues appear in relatively long-duration utterances, while end-to-end ASR systems can correctly transcribe short-duration utterances. In other words, the end-to-end ASR systems have great potential to improve ASR performance if we can address the generation issues.

In this paper, we propose to leverage the end-to-end ASR systems for rescoring hypotheses generated by the DNN-HMM hybrid ASR systems. To this end, we regard the end-to-end ASR systems as neural speech-to-text language models (NS2TLMs), which are language models conditioned on the input acoustic features. We then use these models for rescoring. This removes the generation issues in the end-to-end ASR systems because DNN-HMM hybrid ASR systems can generate speech-aligned hypotheses. In other words, we expect that ASR performance of long-duration utterances will be improved by eliminating the generation issues. Furthermore, the NS2TLMs can be positioned as RNNLMs with rich auxiliary features extracted from the input speech, so we can also improve ASR performance of short-duration utterances by leveraging valid properties of the NS2TLMs.

We use a Japanese ASR lecture task of the Corpus of Spontaneous Japanese (CSJ) [15] to conduct the evaluation. We verify that the NS2TLMs yield better ASR performance than state-of-the-art DNN-HMM hybrid ASR systems with

convolutional and long short-term memory recurrent neural network-based acoustic models. Furthermore, we investigate the relationship between ASR performance and the length of utterances.

This paper is organized as follows. Section II describes related work. Section III explains the RNNLMs, as our proposed method is based on the RNNLM rescoring. Section IV gives the details of the NS2TLMs. Experiments are shown in Section V and the analysis of the experimental results is written in Section VI. Section VII concludes the paper.

II. Related Work

NS2TLMs are based on end-to-end ASR systems [10]–[14] which directly estimate the text from the input speech. They are conditional language models conditioned by the input speech. End-to-end ASR systems based on attentional encoder-decoder were reported to outperform systems based on the connectionist temporal classification (CTC) [16], so we focus on the encoder-decoder based systems in this study. End-to-end ASR systems and DNN-HMM hybrid ASR systems have been studied separately in previous works. In this study, we try to combine the DNN-HMM hybrid ASR system with the end-to-end ASR system to rescore the hypotheses generated by DNN-HMM hybrid ASR system.

NS2TLMs are related to neural language models. RNNLMs in particular [8], [9] have been shown to improve significantly ASR performance. RNNLMs can efficiently capture long-term dependencies of words by embedding long-term contexts into hidden representations in RNNs. Furthermore, LSTM-RNNLMs (LSTMLMs) [17] can further enhance ASR performance due to their ability to reduce the vanishing gradient problem [18]. NS2TLMs estimate probability distributions which are conditioned not only on hidden representations of the word contexts but also on hidden representations composed of the input acoustic features.

RNNLMs conditioned by additional information improve language modeling and speech recognition performance. In [19], latent Dirichlet allocation-based feature extracted from the text is used for additional input in RNNLMs. Bag-of-words representation is used with the same motivation [20]. Shi et al. used part-of-speech tags and conversation related information [21]. In addition, it is reported that integrating acoustic features for input in RNNLMs improves ASR performance [22]–[24]. Prosody features, such as fundamental frequency and pitch, are used for RNNLMs [22], [24]. In [23], global acoustic feature OpenSMILE and i-vector extracted from speech are used for RNNLMs adaptation. NS2TLM is also one of the conditional language models and can directly use richer acoustic information than the other language models.

III. Recurrent Neural Network Based Language Models

This section describes RNNLMs [8]. Fig. 1 shows an example of a RNNLM. RNNLMs estimate generative probabilities

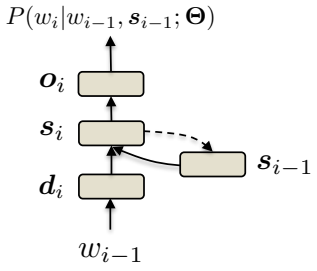

Fig. 1. Example of recurrent neural network language model.

of $\boldsymbol{w} = \{w_1, w_2, \cdots, w_i, \cdots, w_I\}$ as

$$P(\boldsymbol{w}|\boldsymbol{\Theta}) = \prod_{i=1}^{I} P(w_i|w_{i-1}, \boldsymbol{s}_{i-1}; \boldsymbol{\Theta}), \tag{1}$$

where $\boldsymbol{\Theta}$ represents the model parameter. In RNNLMs, each word w_i is mapped to 1-of-K representation and embedded in distributed representation by affine transformation as

$$\boldsymbol{d_i} = \texttt{EMBED}(w_i, \theta_d), \tag{2}$$

where $\texttt{EMBED}(\cdot)$ is a function that converts a word into a distributed representation and θ_d is a trainable parameter. The hidden state is calculated by nonlinear activation function $f(\cdot)$ as

$$\boldsymbol{s_i} = f(\boldsymbol{d}_{i-1}, \boldsymbol{s}_{i-1}, \theta_s). \tag{3}$$

where $\boldsymbol{s_i}$ is the hidden state in the decoder and θ_s is the trainable parameter. Finally, the decoder estimates the word probability in a target hypothesis with a conditional probability as

$$P(w_i|w_{i-1}, \boldsymbol{s}_{i-1}; \boldsymbol{\Theta}) = \texttt{SOFTMAX}(\boldsymbol{s}_i, \theta_o), \tag{4}$$

where θ_o is the trainable parameter.

IV. Neural Speech-to-Text Language Models

In this section, we explain our approach in which end-to-end ASR systems are used as language models. NS2TLMs are based on attentional encoder-decoder models which read variable length inputs in the encoder and predict variable length outputs in the decoder. NS2TLMs are part of the RNNLMs with an additional acoustic feature.

A. Modeling

Figure 2 illustrates an example of NS2TLM. We use a bi-directional LSTM with an attention mechanism [25], [26] as an encoder and a uni-directional LSTM as a decoder. Given the acoustic feature sequence $\boldsymbol{x} = \{x_1, x_2, \cdots, x_j, \cdots, x_J\}$, NS2TLMs estimate the generative probability of $\boldsymbol{w} = \{w_1, w_2, \cdots, w_i, \cdots, w_I\}$ as

$$P(\boldsymbol{w}|\boldsymbol{x}; \boldsymbol{\Lambda}) = \prod_{i=1}^{I} P(w_i|w_{i-1}, \boldsymbol{s}_{i-1}, \bar{\boldsymbol{s}}_i; \boldsymbol{\Lambda}), \tag{5}$$

where $\boldsymbol{\Lambda}$ represents the model parameters.

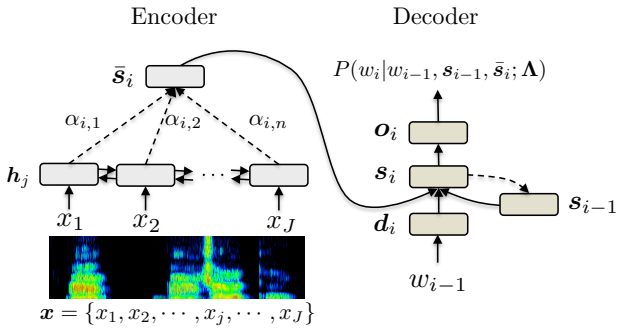

Fig. 2. Example of neural speech-to-text language models.

In the NS2TLM, the acoustic feature is input in the encoder based on bi-directional LSTM as

$$\overrightarrow{\boldsymbol{h}}_j = \overrightarrow{\texttt{LSTM}}(\boldsymbol{x}_j, \overrightarrow{\boldsymbol{h}}_{j-1}, \lambda_{lf}), \quad (6)$$

$$\overleftarrow{\boldsymbol{h}}_j = \overleftarrow{\texttt{LSTM}}(\boldsymbol{x}_j, \overleftarrow{\boldsymbol{h}}_{j+1}, \lambda_{lb}), \quad (7)$$

where $\overrightarrow{\texttt{LSTM}}(\cdot)$ and $\overleftarrow{\texttt{LSTM}}(\cdot)$ represent LSTM functions of forward and backward LSTM. $\lambda_l f$ and $\lambda_l b$ are the trainable model parameters. The encoder hidden state $\boldsymbol{h_j}$ is calculated by concatenating $\overrightarrow{h}_j$ and $\overleftarrow{h}_j$ as

$$\boldsymbol{h_j} = [\overrightarrow{\boldsymbol{h}}_j^\top, \overleftarrow{\boldsymbol{h}}_j^\top]^\top. \quad (8)$$

The context vector $\bar{\boldsymbol{s}}_i$ is constructed in each time-step when estimating generative word probabilities in the decoder as

$$\bar{\boldsymbol{s}}_i = \sum_{j=1}^{J} \alpha_{j,i} \boldsymbol{h}_j, \quad (9)$$

where $\alpha_{j,i}$ is calculated as

$$\alpha_{j,i} = \frac{\exp(\boldsymbol{e_{j,i}})}{\sum_{j=1}^{J} \exp(\boldsymbol{e}_{j,i})}, \quad (10)$$

where $e_{j,i}$ is calculated previous α and matrix F as

$$\boldsymbol{f}_j = \boldsymbol{F} * \boldsymbol{\alpha}_{j-1}, \quad (11)$$

$$\boldsymbol{e}_{j,i} = \texttt{tanh}(\boldsymbol{s}_i, \boldsymbol{h}_j, \boldsymbol{f}_{j,i}, \lambda_e), \quad (12)$$

where $\boldsymbol{s_i}$ is the hidden state in the decoder, "$\cdot$" indicates the dot product function and F and λ_e are the trainable model parameters. In the decoder, the distributed representation $\boldsymbol{d}_{i-1}$ is calculated by the weight matrix as

$$\boldsymbol{d}_{i-1} = \texttt{EMBED}(w_{i-1}, \lambda_d). \quad (13)$$

The hidden state in the decoder is calculated by LSTM function as

$$\boldsymbol{s_i} = \texttt{LSTM}([\boldsymbol{d}_{i-1}, \bar{\boldsymbol{s}}_{i-1}], \boldsymbol{s}_{i-1}, \lambda_s). \quad (14)$$

Then, $\boldsymbol{o}_j$ is calculated by concatenating the decoder hidden state with a context vector as and the hyperbolic tangent function as

$$\boldsymbol{o}_i = \texttt{tanh}([\boldsymbol{s_i}, \bar{\boldsymbol{s}}_i]^\top, \lambda_t), \quad (15)$$

TABLE I
DETAILS OF DATA FOR TRAINING, DEVELOPMENT AND TEST

Data		# of characters	# of words	Hours
Training		12,573,004	7,798,998	644.84
Development		110,616	68,315	5.67
Test	Task1	45,169	27,651	2.28
	Task2	44,915	28,424	2.42
	Task3	29,610	18,238	1.71

where $\boldsymbol{s_i}$ is the hidden state in the decoder, $\bar{\boldsymbol{s}}_i$ denotes the context vector generated from the input acoustic features. Finally, the decoder estimates the word probability in the target hypothesis with a conditional probability as

$$P(w_i|w_{i-1}, \boldsymbol{s}_{i-1}, \bar{\boldsymbol{s}}_i; \boldsymbol{\Lambda}) = \texttt{SOFTMAX}(\boldsymbol{o_j}, \lambda_o). \quad (16)$$

where λ_o is the trainable model parameter.

The trainable model parameters $\boldsymbol{\Lambda} = \{\lambda_{lf}, \lambda_{lb}, \boldsymbol{F}, \lambda_e, \lambda_d, \lambda_s, \lambda_t, \lambda_o\}$ in a NS2TLMs are updated to maximize conditional generative probabilities of transcriptions in the decoder when the acoustic feature is given as a context in the encoder. Thus, the model parameters are updated with minimizing cross entropy loss function:

$$\mathcal{L}(\boldsymbol{\Lambda}) = -\sum_{(\boldsymbol{x}', \boldsymbol{w}') \in \mathcal{D}} \log P(\boldsymbol{w}'|\boldsymbol{x}'; \boldsymbol{\Lambda}), \quad (17)$$

where $\mathcal{D}$ represents pairs of the input speech and manual transcriptions. The training data $\mathcal{D}$ is described as

$$\mathcal{D} = \{(\boldsymbol{x}_1, \boldsymbol{w}_1), (\boldsymbol{x}_2, \boldsymbol{w}_2), \ \cdots, (\boldsymbol{x}_N, \boldsymbol{w}_N)\}. \quad (18)$$

B. Rescoring Hypotheses

NS2TLMs are utilized for rescoring ASR hypotheses generated from DNN-HMM hybrid ASR systems. The ASR score calculated by DNN-HMM hybrid ASR system is linearly interpolated with a log generative probability obtained by NS2TLMs. Given the acoustic feature sequence $\boldsymbol{x} = \{x_1, x_2, \cdots, x_j, \cdots, x_J\}$, 1-best ASR result $\hat{\boldsymbol{w}}$ is determined by

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \{\beta \log P(\boldsymbol{w}|\boldsymbol{x}; \boldsymbol{\Lambda}) + (1-\beta) \log P(\boldsymbol{w}|\boldsymbol{x}; \boldsymbol{\eta})\}, \quad (19)$$

where $P(\boldsymbol{w}|\boldsymbol{x}; \boldsymbol{\eta})$ denotes the ASR score calculated by DNN-HMM hybrid ASR system with their parameters $\boldsymbol{\eta}$ and β is the interpolation weight of NS2TLM.

V. EXPERIMENTS

A. Setups

All experiments were performed on CSJ [15], which is a Japanese lecture corpus. Table I shows the details of the data for training, development and test. End-to-end and DNN-HMM ASR system were trained with 40 mel-scale filter-bank features.

TABLE II
%CERs ON THREE CSJ EVALUATION SETS IN DIFFERENT MODELS

Model	Task1	Task2	Task3	AVG.
End-to-end system (NS2TLM)	13.49	9.96	12.25	11.90
DNN-HMM hybrid system	11.85	9.25	9.84	10.31
+ LSTMLM	11.12	8.49	9.21	9.61
+ NS2TLM	10.38	7.56	8.24	8.73
+ LSTMLM + NS2TLM	**10.20**	**7.38**	**7.96**	**8.51**

We prepared an acoustic-to-character based end-to-end ASR system (NS2TLM). The end-to-end ASR systems had bi-directional LSTM with 4 hidden layers and 320 units in each layer and direction in the encoder and uni-directional LSTM with 1 hidden layer and 320 LSTM units in the decoder. The vocabulary size was 3251 symbols corresponding to the dimensions of the output target. The beam size was set to 20 for the beam search decoding and the candidates hypotheses were re-ranked based on the length normalized scores [27].

We used a CNN-LSTM acoustic model in DNN-HMM the hybrid ASR system. In the CNN-LSTM acoustic model, each static and dynamic component was sliced within 11 frames, which was composed as 3 feature maps. We used 1 convolutional layer with 128 features maps in which 5×11 frequency-time filters. For pooling, 2×1 frequency-time max pooling was performed. In addition, CNN output was fed into 2 LSTM layers, each of which had 1024 cells. LSTM output was fed into a softmax layer. We prepared a 3-gram language model. The speech recognizer included a weighted finite state transducer based decoder [28].

LSTMLM had 2 hidden layers and 520 LSTM units in each layer. The vocabulary size was 67780 words corresponding to the dimensions of the input and output. The dropout ratio was set to 0.3 in each hidden layer.

We used the 100-best list generated from each utterance to rescore with NS2TLMs and LSTMLMs. The DNN-HMM hybrid ASR system was used for the hypotheses generation in all experiments. The NS2TLM score was interpolated with the ASR score in accordance with Eq. (19). In the case of using LSTMLM, the score was interpolated with the 3-gram language model score.

B. Results

Table II reports the character error rates (CERs) in three CSJ evaluation sets. The DNN-HMM hybrid ASR system shows lower CER than the attention-based end-to-end ASR system in these setups. The NS2TLM gave larger CER reduction than the LSTMLM rescoring in all evaluation sets. Acoustic features can help achieving significant improvement in ASR performance. In addition, a combination of NS2TLM and LSTMLM shows further improvement, achieving best CER of 8.73% on average in all three tasks in our experiments.

We show the relationship between the additional language model weights of NS2TLM and LSMTLM and the CER in Fig. 3. NS2TLM shows the lowest CER compared with other systems even when the additional language model weight was set to large values of 0.5-1.0.

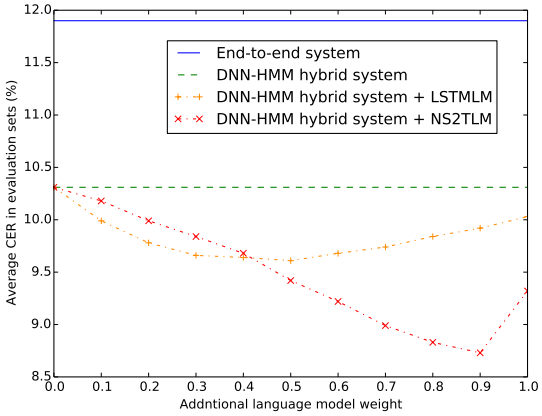

Fig. 3. Average CER in evaluation sets and additional language model weight

VI. ANALYSIS

We analyzed the effects of NS2TLMs from the point of view of the length of utterances. Figure 4 demonstrates the relationship between the reduction of CER and the length of utterances: orange plots represent CER differences between the end-to-end ASR system and the DNN-HMM hybrid ASR system; blue plots represent CER differences between the DNN-HMM hybrid ASR system and rescored by NS2TLM (DNN-HMM hybrid system + NS2TLM) with respect to the number of characters in utterances. The orange plots show that the end-to-end ASR systems were inferior to the hybrid systems when handling long-duration utterances but yielding comparative performance when handling short-duration utterances with a number of characters 20 or less. This indicates that the end-to-end ASR systems have difficulty dealing with long-duration utterances. In addition, the blue plots show that the DNN-HMM hybrid ASR systems with NS2TLM attained comparatively higher performance when handling short-duration utterances than when handling long-duration utterances. This is because the NS2TLM was good at capturing short-duration utterances. These results confirm that our proposed method is effective in compensating for weaknesses and in leveraging the strength of the end-to-end ASR systems.

VII. CONCLUSIONS

In this paper, we proposed to leverage end-to-end ASR systems for assisting DNN-HMM hybrid ASR systems. Experimental results showed that NS2TLM rescoring gave larger CER reduction than LSTMLM rescoring in a Japanese lecture task. The best CER was obtained by rescoring with a combination of NS2TLM and LSTMLM scores, which reduces CER of the state-of-the-art DNN-HMM hybrid ASR system including CNN and LSTM acoustic model by 17.5%. Analysis revealed a relationship between ASR performance and utterance length in end-to-end ASR systems, DNN-HMM hybrid ASR systems and NS2TLMs rescored systems. Rescoring by NS2TLMs is

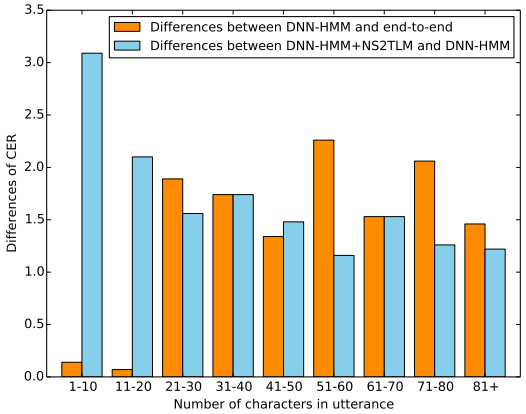

Fig. 4. Relationship between the number of characters in an utterance and CER differences.

highly effective for short-duration utterances because end-to-end ASR systems perform better than DNN-HMM hybrid ASR systems for short-duration utterances.

REFERENCES

[1] "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[2] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," *In Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4277–4280, 2012.

[3] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR."

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

[5] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," pp. 273–278, 2013.

[6] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[7] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[8] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1045–1048, 2010.

[9] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, "Recurrent neural network based language modeling in meeting recognition," *In proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2877–2880, 2011.

[10] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," *In Proc. International Conference on Machine Learning, (ICML)*, pp. 1764–1772, 2014.

[11] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," vol. abs/1412.5567, 2014.

[12] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: end-to-end speech recognition using deep RNN models and wfst-based decoding," *In Proc. Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 167–174, 2015.

[13] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *In Proc. Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 577–585, 2015.

[14] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," *In Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4945–4949, 2016.

[15] S. Furui, K. Maekawa, and H. Isahara, "A Japanese national project on spontaneous speech corpus and processing technology," *In Proc. ASR2000 - Automatic Speech Recognition: Challenges for the new Millenium*, pp. 244–248, 2000.

[16] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *In Proc. International Conference on Machine Learning (ICML)*, pp. 369–376, 2006.

[17] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 194–197, 2012.

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

[19] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," *In Proc. Spoken Language Technology Workshop (SLT)*, pp. 234–239, 2012.

[20] K. Irie, R. Schlüter, and H. Ney, "Bag-of-words input for long history representation in neural network-based language models for speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2371–2375, 2015.

[21] Y. Shi, P. Wiggers, and C. M. Jonker, "Towards recurrent neural networks language models with linguistic and contextual features," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1664–1667, 2012.

[22] T. Fu, Y. Han, X. Li, Y. Liu, and X. Wu, "Integrating prosodic information into recurrent neural network language model for speech recognition," *In Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1194–1197, 2015.

[23] S. Toyama, D. Saito, and N. Minematsu, "Use of global and acoustic features associated with contextual factors to adapt language models for spontaneous speech recognition," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 543–547, 2017.

[24] M. Hentschel, A. Ogawa, M. Delcroix, T. Nakatani, and Y. Matsumoto, "Exploiting imbalanced textual and acoustic data for training prosodically-enhanced rnnlms," 2017, pp. 618–621.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv:1409.0473*, 2014.

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *In Proc. Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 3104–3112, 2014.

[27] F. Cromierès, C. Chu, T. Nakazawa, and S. Kurohashi, "Kyoto university participation to WAT," *In Proc. the 3rd Workshop on Asian Translation (WAT)*, pp. 166–174, 2016.

[28] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient wfst-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Trans. Audio, Speech & Language Processing*, vol. 15, no. 4, pp. 1352–1365, 2007.