Interspeech 2018
2-6 September 2018, Hyderabad

Neural Error Corrective Language Models for Automatic Speech Recognition

Tomohiro Tanaka, Ryo Masumura, Hirokazu Masataki, Yushi Aono

NTT Media Intelligence Laboratories, NTT Corporation

{t.tomohiro, masumura.ryo, masataki.hirokazu, aono.yushi}@lab.ntt.co.jp

Abstract

We present novel neural network based language models that can correct automatic speech recognition (ASR) errors by using speech recognizer output as a context. These models, called neural error corrective language models (NECLMs), utilizes ASR hypotheses of a target utterance as a context for estimating the generative probability of words. NECLMs are expressed as conditional generative models composed of an encoder network and a decoder network. In the models, the encoder network constructs context vectors from N-best lists and ASR confidence scores generated in a speech recognizer. The decoder network rescores recognition hypotheses by computing a generative probability of words using the context vectors so as to correct ASR errors. We evaluate the proposed models in Japanese lecture ASR tasks. Experimental results show that NECLM achieve better ASR performance than a state-of-the-art ASR system that incorporate a convolutional neural network acoustic model and a long short-term memory recurrent neural network language model.

Index Terms: automatic speech recognition, language models, speech recognition error correction, conditional generative models

1. Introduction

Language models are one of the essential components in automatic speech recognition (ASR) systems. Their role is to estimate generative probabilities of output strings generated from acoustic models or other speech recognizers. In state-of-the-art ASR systems, two language models are often introduced into two-pass decoding. In the first decoding pass, hypotheses are generated using n-gram language models which are compatible with decoding algorithms. In the second decoding pass, neural network based language models (NNLMs) [1, 2] are applied for rescoring the hypotheses. It is known that NNLMs improve ASR performance by combining with n-gram language models. Among NNLMs, recurrent neural network based language models (RNNLMs) [3, 4] have been shown to significantly improve ASR performance because they can capture longer-term contexts of word sequences. Furthermore, long short-term memory RNNLMs (LSTMLMs) [5] can further enhance ASR performance due to their ability to help reduce the vanishing gradient problem [6].

Most language models including n-grams and RNNLMs are usually built on the basis of correct word sequences such as manual transcriptions. On the other hand, language models that utilize ASR hypotheses have also been proposed to take account of ASR error tendencies. Discriminative language models (DLMs) [7–10], which train functions with a discriminative criterion, are the most famous language models for capturing ASR error tendencies. In addition, the discriminative criterion is introduced into NNLMs [11, 12]. The discriminative criterion is effective for reducing ASR errors. However, previous language models cannot consider how errors tend to occur in a target utterance because they do not utilize ASR hypotheses for computing generative probabilities of words at all.

This paper proposes neural error corrective language models (NECLMs), which directly utilize ASR hypotheses as contexts for estimating generative probabilities of words. The NECLMs assume that ASR hypotheses have already been generated by a speech recognizer, while previous LMs do not need them. In order to handle the ASR hypotheses, the NECLMs are composed as conditional generative models using an encoder network and a decoder network. In the encoder network, ASR hypotheses of a target utterance are encoded to continuous representations. The decoder network can compute generative probabilities of a word sequence by leveraging the continuous representations. Therefore, the NECLMs can directly reduce ASR errors of the target utterance. In order to accurately capture the ASR error tendencies of the target utterance, we present several modeling methods that utilize multiple ASR hypotheses and confidence measures obtained by a speech recognizer. We also present several training methods to reduce a lot of ASR errors.

We evaluate NECLMs on a Japanese ASR lecture task of the Corpus of Spontaneous Japanese (CSJ) [13]. The results verify that NECLMs provide better ASR performance than state-of-the-art ASR setups.

This paper is organized as follows. Section 2 describes related work. Section 3 defines error corrective language modeling. Section 4 details NECLMs. Experiments are shown in Section 5 and Section 6 concludes the paper with a summary and a mention of future work.

2. Related Work

2.1. End-to-End ASR

Our proposed models are related to an end-to-end approach for ASR [14–18]. A sequence-to-sequence based end-to-end ASR system was reported to outperform a connectionist temporal classification based system. Whereas the sequence-to-sequence based end-to-end model for ASR is a conditional generative model for input speech, our proposed models are conditional generative models of ASR result. Speech recognizers need to exist before NECLMs can correct errors. Unlike the end-to-end ASR, NECLMs capture speech recognition error tendencies directly. In the work reported in this paper, conventional speech recognition is conducted before using NECLMs. It is possible that the end-to-end ASR can be used instead of a conventional speech recognizer.

2.2. Unsupervised Language Model Adaptation

NECLMs are similar to unsupervised language model adaptation methods [19], which calculate the generative probability of words by using speech recognition results [20–22]. Unsupervised language model adaptation methods utilize correct

10.21437/Interspeech.2018-1430

word sequence, but not ASR hypotheses including ASR errors. Unlike unsupervised language model adaptation methods, NECLMs utilize ASR hypotheses to directly model ASR errors.

2.3. Neural Post Editing for Machine Translation

In machine translation, neural network based post editing (NPE) methods have been proposed and shown to significantly improve machine translation in both objective and subjective evaluations [23, 24]. An NPE model is trained with pairs of a translation result and its correct sentence. Pal et al. [23] define NPE as a post-processing module. Our proposed models for ASR are also assumed to be post-processing modules, but we formulate them as language models that utilize ASR hypotheses as a context.

3. Error Corrective Language Models

This section details the definition of error corrective language models (ECLMs). The ECLMs are language models that use speech recognizer output as a context for estimating generative probabilities of words. Given speech recognizer output $\boldsymbol{H}(\boldsymbol{x};\theta)$ with input speech $\boldsymbol{x}$, the generative probability of an input word sequence $\boldsymbol{w} = \{w_1, w_2, \cdots, w_i, \cdots, w_I\}$ is written as

$$P(\boldsymbol{w}|\boldsymbol{x}) = P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda}), \tag{1}$$

where θ denotes parameters in a speech recognizer and $\boldsymbol{\Lambda}$ represents parameters of ECLMs. The speech recognizer includes acoustic models and typical language models such as n-grams and LSTMLMs. Given ASR hypotheses as an N-best list, $\boldsymbol{H}(\boldsymbol{x};\theta)$ can be written as

$$\begin{aligned}\boldsymbol{H}(\boldsymbol{x};\theta) = \{&(\boldsymbol{r}^1, P(\boldsymbol{r}^1), P(\boldsymbol{x}|\boldsymbol{r}^1)), \cdots, \\ &(\boldsymbol{r}^n, P(\boldsymbol{r}^n), P(\boldsymbol{x}|\boldsymbol{r}^n)), \cdots, (\boldsymbol{r}^N, P(\boldsymbol{r}^N), P(\boldsymbol{x}|\boldsymbol{r}^N))\},\end{aligned} \tag{2}$$

where $\boldsymbol{r}^n$ is the hypothesis that has the n-th highest ASR score in an N-best list, $P((\boldsymbol{x}|\boldsymbol{r}^n)))$ is the generative probability of x computed by the acoustic models and $P(\boldsymbol{r}^n)$ is the generative probability of $\boldsymbol{r}^n$ computed by the typical language models. Unlike typical language models, ECLMs are trained from both outputs of a speech recognizer and corresponding transcriptions. Thus, $\boldsymbol{\Lambda}$ are trained to correct ASR errors that occurred by θ.

ECLMs are utilized for rescoring ASR hypotheses. In the rescoring, the ASR score calculated by the speech recognizer is linearly interpolated with log generative probability obtained by an ECLM. The 1-best ASR result $\hat{\boldsymbol{w}}$ is determined by

$$\begin{aligned}\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \{&\beta \log P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda}) \\ &+ (1-\beta)\log P(\boldsymbol{w}|\boldsymbol{x};\theta)\},\end{aligned} \tag{3}$$

where $P(\boldsymbol{w}|\boldsymbol{x};\theta)$ denotes the ASR score calculated by the speech recognizer and β is the interpolation weight of the ECLM.

4. Neural Error Corrective Language Models

4.1. Model Definition

NECLMs are a type of ECLM based on a sequence-to-sequence model. Given sequences $\boldsymbol{X}$ and $\boldsymbol{Y}$, the sequence-to-sequence model can calculate the conditional generative probability of $P(\boldsymbol{X}|\boldsymbol{Y})$. NECLMs calculate the conditional generative probability $P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda})$ to use this ability. In the NECLMs, the hypotheses used to construct contexts are fixed. Then, the generative probabilities of each hypothesis in the N-best list generated from the speech recognizer are estimated in the decoder. We can select hypotheses from the N-best list to construct the encoder context. In ths study, we examine both single and multiple hypotheses scoring.

4.1.1. Single Hypothesis Based NECLMs

In single hypothesis based NECLMs, only a single hypothesis is utilized to construct the contexts. In this case, $P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda})$ is formulated as:

$$P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda}) = P(\boldsymbol{w}|\boldsymbol{r}^n;\boldsymbol{\Lambda}). \tag{4}$$

We examine two strategies in this study. One is to use the hypothesis with the highest ASR score (*single-hyp. highest ASR*). The other is to use the hypothesis with the lowest ASR score (*single-hyp. lowest ASR*).

4.1.2. Multiple Hypotheses Based NECLMs

Multiple hypotheses based NECLMs utilize the top K hypotheses in the N-best result. To this end, we utilize a posterior probability of $\boldsymbol{r}^n$ given $\boldsymbol{x}$ for weighting each hypothesis. If we assume that each hypothesis has the same posterior probability, $P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda})$ can be written as

$$P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda}) = \sum_{k=1}^{K} P(\boldsymbol{r}^k|\boldsymbol{x}) P(\boldsymbol{w}|\boldsymbol{r}^k;\boldsymbol{\Lambda}), \tag{5}$$

$$\approx \frac{1}{K}\sum_{k=1}^{K} P(\boldsymbol{w}|\boldsymbol{r}^k;\boldsymbol{\Lambda}), \tag{6}$$

where K represents the number of hypotheses used to construct encoder contexts in scoring. We call this scoring *multi-hyp. average*. On the other hand, if we assume that the posterior probabilities correspond to the confidence scores of the speech recognizer, $P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda})$ can be formulated as

$$P(\boldsymbol{w}|\boldsymbol{H}(\boldsymbol{x};\theta);\boldsymbol{\Lambda}) = \sum_{k=1}^{K} P(\boldsymbol{r}^k|\boldsymbol{x}) P(\boldsymbol{w}|\boldsymbol{r}^k;\boldsymbol{\Lambda}), \tag{7}$$

$$\approx \sum_{k=1}^{K} \frac{P(\boldsymbol{x}|\boldsymbol{r}^k)P(\boldsymbol{r}^k)}{\sum_{l=1}^{N} P(\boldsymbol{x}|\boldsymbol{r}^l)P(\boldsymbol{r}^l)} P(\boldsymbol{w}|\boldsymbol{r}^k;\boldsymbol{\Lambda}). \tag{8}$$

We call this scoring *multi-hyp. confidence*.

4.2. Neural Network Based Modeling

We next explain how to calculate $P(\boldsymbol{w}|\boldsymbol{r};\boldsymbol{\Lambda})$ in both single hypothesis and multiple hypotheses scoring. NECLMs estimate generative probability $P(\boldsymbol{w}|\boldsymbol{r};\boldsymbol{\Lambda})$ by a neural network. Figure 1 illustrates an example of NECLM whose encoder context is single ASR hypothesis. We use a bi-directional LSTM (bi-LSTM) with attention mechanism [25, 26] as an encoder and a uni-directional LSTM (uni-LSTM) as a decoder. Given an ASR hypothesis $\boldsymbol{r}$ and a scoring target word sequence $\boldsymbol{w} = \{w_1, w_2, \cdots, w_i, \cdots, w_I\}$, the conditional generative probability is calculated as

$$P(\boldsymbol{w}|\boldsymbol{r};\boldsymbol{\Lambda}) = \prod_{i=1}^{I} P(w_i|w_{i-1}, \boldsymbol{s}_{i-1}, \bar{\boldsymbol{s}}_i;\boldsymbol{\Lambda}), \tag{9}$$

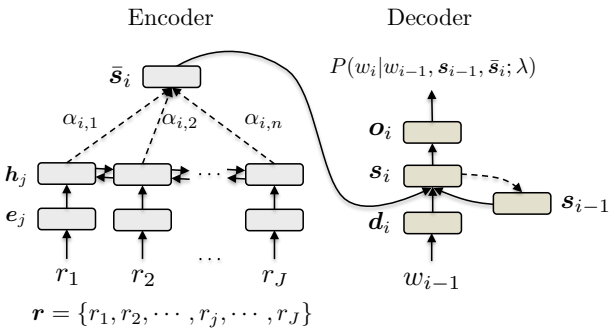

Figure 1: *Neural error corrective language model with single hypothesis.*

where $\boldsymbol{s}_{i-1}$ denotes a hidden state in the LSTM decoder, $\bar{\boldsymbol{s}}_i$ is the context vector of an ASR hypothesis in the encoder and $\boldsymbol{\Lambda}$ is the trainable parameters in the NECLM. In the encoder, a word in an ASR hypothesis r_j is mapped to 1-of-K representation and embedded in distributed representation by affine transformation as

$$\boldsymbol{e_j} = \texttt{EMBED}(r_j, \lambda_e), \tag{10}$$

where $\texttt{EMBED}(\cdot)$ is the function that converts from a word to a distributed representation and λ_e is the trainable parameter. Hidden states in forward and backward LSTMs are calculated as

$$\overrightarrow{\boldsymbol{h}}_j = \overrightarrow{\texttt{LSTM}}(\boldsymbol{e}_j, \overrightarrow{\boldsymbol{h}}_{j-1}, \lambda_{lf}), \tag{11}$$

$$\overleftarrow{\boldsymbol{h}}_j = \overleftarrow{\texttt{LSTM}}(\boldsymbol{e}_j, \overleftarrow{\boldsymbol{h}}_{j+1}, \lambda_{lb}), \tag{12}$$

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

$$\boldsymbol{h_j} = [\overrightarrow{\boldsymbol{h}}_j^\top, \overleftarrow{\boldsymbol{h}}_j^\top]^\top. \tag{13}$$

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

$$\bar{\boldsymbol{s}}_i = \sum_{j=1}^{J} \alpha_{j,i} \boldsymbol{h}_j, \tag{14}$$

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

$$\alpha_{j,i} = \frac{\exp(\boldsymbol{s}_i \cdot \boldsymbol{h}_j)}{\sum_{l=1}^{J} \exp(\boldsymbol{s}_i \cdot \boldsymbol{h_l})}, \tag{15}$$

where $\boldsymbol{s_i}$ is the hidden state in the decoder and "$\cdot$" indicates a dot product function. In the decoder, distributed representation $\boldsymbol{d}_{i-1}$ is calculated by a different weight matrix from the encoder as

$$\boldsymbol{d}_{i-1} = \texttt{EMBED}(w_{i-1}, \lambda_d), \tag{16}$$

where λ_d is the trainable parameter. The hidden state in the decoder is calculated by the LSTM function as

$$\boldsymbol{s_i} = \texttt{LSTM}([\boldsymbol{d}_{i-1}, \bar{\boldsymbol{s}}_{i-1}], \boldsymbol{s}_{i-1}, \lambda_s), \tag{17}$$

Table 1: *Datasets for NECLMs*

Data	# of words	Hours
Training	5,209,203	545.41
Development	517,480	54.21
Test	64,166	6.42

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

$$\boldsymbol{o}_i = \texttt{tanh}([\boldsymbol{s_i}, \bar{\boldsymbol{s}}_i]^\top, \lambda_t), \tag{18}$$

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

$$P(w_i|w_{i-1}, \boldsymbol{s}_{i-1}, \bar{\boldsymbol{s}}_i, \boldsymbol{\Lambda}) = \texttt{SOFTMAX}(\boldsymbol{o_j}, \lambda_o). \tag{19}$$

where λ_o is the trainable parameter. By repeating the calculation, the NECLMs can estimate generative probability $P(\boldsymbol{w}|\boldsymbol{r}; \boldsymbol{\Lambda})$.

4.3. Training

Model parameters in the NECLMs are updated to maximize the conditional generative probability of manual transcription in the decoder when giving an ASR hypothesis as a context in the encoder. Thus, the model parameters are optimized by minimizing cross entropy loss function:

$$\mathcal{L}(\boldsymbol{\Lambda}) = -\sum_{(\boldsymbol{r}', \boldsymbol{w}') \in \mathcal{D}} \log P(\boldsymbol{w}'|\boldsymbol{r}'; \boldsymbol{\Lambda}), \tag{20}$$

where $\mathcal{D}$ is pairs of ASR results and manual transcriptions. In training the NECLMs with all hypotheses in an N-best list, training data $\mathcal{D}$ is described as

$$\mathcal{D} = \{(\boldsymbol{r}_1, \boldsymbol{w}_1), (\boldsymbol{r}_2, \boldsymbol{w}_2), \ \cdots, (\boldsymbol{r}_N, \boldsymbol{w}_N)\}. \tag{21}$$

For encoder context, we investigate two hypotheses selected from N-best list in training. One has the highest ASR score (*high ASR*) and the other has the highest WER (*high WER*). We expect the NECLMs will be able to model various ASR errors by using *high WER*.

5. Experiments

5.1. Experimental Setups

We evaluated the NECLMs with a Japanese lecture ASR task of the CSJ. Datasets for them are shown in Table 1. The vocabulary size of their training data was 78,688 words. The training set for a first pass N-gram language model and an LSTMLM was 6,100,688 words in CSJ lectures, which had a vocabulary size of 78,964 words. An acoustic model was trained with approximately 1,000 hours of speech including the CSJ and other private data. We evaluated the proposed models by averaging WERs over three standard CSJ evaluation sets.

The baseline system uses a convolutional neural network based acoustic model. The language model in the first pass decoding is a Kneser-Ney smoothed 3-gram model [27]. The LSTMLM has 2 layers and 512 units in each hidden layer. The speech recognizer includes a weighted finite state transducer based decoder [28].

Table 2: *Encoder context hypotheses and WERs on evaluation sets*

Encoder context hypotheses		
training	scoring	%WER
high ASR	*single-hyp. high ASR*	20.80
	single-hyp. low ASR	21.57
	multi-hyp. average	20.70
	multi-hyp. confidence	20.76
high WER	*single-hyp. high ASR*	20.63
	single-hyp. low ASR	21.38
	multi-hyp. average	**20.43**
	multi-hyp. confidence	**20.43**
No rescoring		21.96

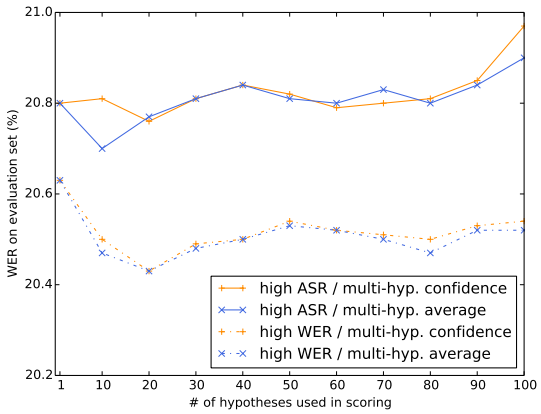

Figure 2: *WERs for different numbers of hypotheses K in calculating generative probability for each calculating method*

An NECLM consists of an encoder network and a decoder network. The encoder network is bi-LSTM with an attention mechanism, which has a 2-layer LSTM with 512 units per layer in the forward and backward directions. The decoder network has a 2-layer uni-LSTM with 512 units in each layer. Output size corresponds to vocabulary size. In NECLM training, development data was used for scheduling the learning rate. The learning rate for all NECLMs in these experiments was set to 0.1 initially and reduced when cross entropy loss was not reduced below that of the previous epoch. The training was stopped when the learning rate reached 0.01. The dropout ratio in each LSTM layer was set to 0.3.

We used the 100-best list generated from each utterance to train NECLMs and evaluate NECLMs and LSTMLMs. The NECLM score was interpolated with the ASR score in accordance with Eq. (3). The interpolation weight β was changed from 0 to 1 in 0.1 step. When using an LSTMLM score, the score is interpolated with the n-gram language model score.

5.2. Results

5.2.1. Comparison with Context Hypotheses

We investigated hypotheses for encoder contexts in NECLMs to effectively model ASR errors. Each hypothesis pair was compared by using WER over the evaluation set.

For the context hypotheses, a single hypothesis and multiple hypotheses were selected in scoring. We use the highest ASR score hypothesis (*single-hyp. high ASR*) and the lowest ASR score (*single-hyp. low ASR*) hypothesis as a single hypothesis. When scoring with multiple hypotheses, the generative probability was calculated in NECLMs in accordance with Eq.(6) as *multi-hyp. average* and Eq. (8) as *multi-hyp. confidence*. For the training, we utilized the highest ASR score hypothesis (*high ASR*) and the highest WER hypothesis(*high WER*).

Table 2 shows the WER performance when using different hypotheses in the encoder as a context. All combinations of context hypotheses in training and scoring of NECLMs provided WER improvement over the baseline system. Training with *high WER* hypothesis showed lower WER than training with *high ASR* hypothesis for the same evaluation context. It is assumed that *high WER* hypotheses enables NECLMs to capture more errors of various types than *high ASR* hypotheses training. The evaluation with *low ASR* hypothesis provided less improvement from *high ASR* hypothesis in evaluation encoder context. In the scoring with multiple hypotheses, averaging scoring as well as confidence scoring showed the best WER 20.43%.

Figure 2 shows WERs for different numbers of hypotheses K when calculating generative probability for each calculating method. Training with *low ASR* or *high ASR* hypothesis and scoring with averaging score calculation shows the lowest WER for all K. As can be seen in the figure, using K = 10-20 is better than using K = 30-100.

Table 3: *WERs in different systems un-rescored and rescored by NECLM*

Model	%WER
baseline	21.96
+ NECLM	20.43
baseline + LSTMLM	20.19
+ NECLM	**19.82**
100-best oracle	14.28

5.2.2. Evaluation with LSTMLM.

Table 3 shows WERs in different systems un-rescored and rescored by an NECLM. In the table, "100-best oracle" represents the lowest WER in a 100-best list. This is the lower-bound of WER in N-best list rescoring. With NECLM rescoring, the WER for the baseline system was reduced by 6.97%. We also achieved 1.83% relative WER reduction by the LSTMLM rescoring system. This indicates that the properties of the NECLM differed from those of the LSTMLM.

6. Conclusions and Future Work

In this paper, we proposed NECLMs, which directly utilizes ASR results as contexts to estimate generative probabilities of words. We defined ECLMs and formulated NECLMs as conditional generative models with neural networks. To model ASR errors effectively, we investigated hypotheses to use construct context in the encoder. In experiments we conducted, lower WER than in other methods was achieved by NECLMs trained by multiple hypotheses with the highest WER that were selected from an N-best list. Specifically, we achieved the relative WER reduction of 1.83% with respect to a system including a convolutional neural network acoustic model and an LSTMLM. Future work includes comparing our proposed models with discriminative language models and extending the encoder context to use multiple hypotheses with a confusion network [29].

7. References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Z. Chen, K. Lee, and M. Li, "Discriminative training on language model," *In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 493–496, 2000.

[8] P. Xu, S. Khudanpur, M. Lehr, E. T. Prud'hommeaux, N. Glenn, D. Karakos, B. Roark, K. Sagae, M. Saraclar, I. Shafran, D. M. Bikel, C. Callison-Burch, Y. Cao, K. B. Hall, E. Hasler, P. Koehn, A. Lopez, M. Post, and D. Riley, "Continuous space discriminative language modeling," *In Proc. International Conference on Acoustics, Speech, & Signal Processing (ICASSP)*, pp. 2129–2132, 2012.

[9] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," *In proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 47–54, 2004.

[10] T. Oba, T. Hori, A. Nakamura, and A. Ito, "Round-robin duel discriminative language models," *IEEE Trans. Audio, Speech & Language Processing*, vol. 20, no. 4, pp. 1244–1255, 2012.

[11] Y. Tachioka and S. Watanabe, "Discriminative method for recurrent neural network language models," *In Proc. International Conference on Acoustics, Speech, & Signal Processing (ICASSP)*, pp. 5386–5390, 2015.

[12] T. Hori, C. Hori, S. Watanabe, and J. R. Hershey, "Minimum word error training of long short-term memory recurrent neural network language models for speech recognition," *In Proc. International Conference on Acoustics, Speech, & Signal Processing (ICASSP)*, pp. 5990–5994, 2016.

[13] 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.

[14] 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.

[15] 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," *arXiv preprint arXiv:1412.5567*, 2014.

[16] 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.

[17] 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.

[18] 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.

[19] J. R. Bellegarda, "Statistical language model adaptation: review and perspectives," *Speech Communication*, vol. 42, no. 1, pp. 93–108, 2004.

[20] R. Gretter and G. Riccardi, "On-line learning of language models with word error probability distributions," *In proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 557–560, 2001.

[21] M. Bacchiani and B. Roark, "Unsupervised language model adaptation," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 224–227, 2003.

[22] L. Chen, J. Gauvain, L. Lamel, and G. Adda, "Unsupervised language model adaptation for broadcast news," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 220–223, 2003.

[23] S. Pal, S. K. Naskar, M. Vela, and J. van Genabith, "A neural network based approach to automatic post-editing," *In Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[24] Q. Liu, S. K. Naskar, J. van Genabith, S. Pal, and M. Vela, "Neural automatic post-editing using prior alignment and reranking," *In Proc. European Chapter of the Association for Computational Linguistics (EACL)*, pp. 349–355, 2017.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint 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] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 181–184, 1995.

[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.

[29] R. Masumura, Y. Ijima, T. Asami, H. Masataki, and R. Higashinaka, "Neural confnet classification: Fully neural network based spoken utterance classification using word confusion networks," *In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6039–6043, 2018.