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 recovers 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 incorporates 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
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 \( H(x; \theta) \) with input speech \( x \), the generative probability of an input word sequence \( w = \{w_1, w_2, \cdots, w_t\} \) is written as

\[
P(w|x) = P(w|H(x; \theta); \Lambda),
\]

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

\[
H(x; \theta) = \{(r^1, P(r^1), P(x|r^1)), \cdots, (r^n, P(r^n), P(x|r^n)), \cdots, (r^N, P(r^N), P(x|r^N))\},
\]

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

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 \( \hat{w} \) is determined by

\[
\hat{w} = \text{arg max}_w \{\beta \log P(w|H(x; \theta); \Lambda) + (1 - \beta) \log P(w|x; \theta)\},
\]

where \( P(w|x; \theta) \) denotes the ASR score calculated by the speech recognizer and \( \beta \) 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 \( X \) and \( Y \), the sequence-to-sequence model can calculate the conditional generative probability of \( P(X|Y) \). NECLMs calculate the conditional generative probability \( P(w|H(x; \theta); \Lambda) \) 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 this 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, \( P(w|H(x; \theta); \Lambda) \) is formulated as

\[
P(w|H(x; \theta); \Lambda) = P(w|r^n; \Lambda),
\]

where \( r^n \) denotes parameters in a speech recognizer and \( \Lambda \) the generative probability of an 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 \( r^n \) given \( x \) for weighting each hypothesis. If we assume that each hypothesis has the same posterior probability, \( P(w|H(x; \theta); \Lambda) \) can be written as

\[
P(w|H(x; \theta); \Lambda) = \sum_{k=1}^{K} P(r^k|x)P(w|r^k; \Lambda),
\]

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, \( P(w|H(x; \theta); \Lambda) \) can be formulated as

\[
P(w|H(x; \theta); \Lambda) = \sum_{k=1}^{K} P(x|r^k)P(w|r^k; \Lambda),
\]

We call this scoring multi-hyp. confidence.

4.2. Neural Network Based Modeling

We next explain how to calculate \( P(w|r; \Lambda) \) in both single hypothesis and multiple hypotheses scoring. NECLMs estimate generative probability \( P(w|r; \Lambda) \) 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 \( r \) and a scoring target word sequence \( w = \{w_1, w_2, \cdots, w_t\} \), the conditional generative probability is calculated as

\[
P(w|r; \Lambda) = \prod_{i=1}^{t} P(w_i|w_{i-1}, s_{i-1}, \bar{s}_i; \Lambda),
\]
where \( s_{i-1} \) denotes a hidden state in the LSTM decoder, \( \bar{s}_i \) is the context vector of an ASR hypothesis in the encoder and \( \Lambda \) is the trainable parameters in the NECLMs. 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

\[
e_j = \text{EMBED}(r_j, \lambda_o),
\]

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

\[
\overrightarrow{h}_j = \text{LSTM}(e_j, \overrightarrow{h}_{j-1}, \lambda_f),
\]

\[
\overleftarrow{h}_j = \text{LSTM}(e_j, \overleftarrow{h}_{j+1}, \lambda_b),
\]

where \( \text{LSTM}(\cdot) \) represent LSTM functions of forward and backward LSTMs in the encoder and \( \lambda_f \) and \( \lambda_b \) are the trainable parameters. The encoder hidden state \( h_j \) is calculated by concatenating \( \overrightarrow{h}_j \) and \( \overleftarrow{h}_j \) as

\[
h_j = [\overrightarrow{h}_j, \overleftarrow{h}_j]^{\top}.
\]

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

\[
\bar{s}_i = \sum_{j=1}^{J} \alpha_{j,i} h_j,
\]

where \( \alpha_{j,i} \) is calculated as

\[
\alpha_{j,i} = \frac{\exp(s_i \cdot h_j)}{\sum_{j'=1}^{J} \exp(s_i \cdot \overrightarrow{h}_{j'})},
\]

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

\[
d_{i-1} = \text{EMBED}(w_{i-1}, \lambda_d),
\]

where \( \lambda_d \) is the trainable parameter. The hidden state in the decoder is calculated by the LSTM function as

\[
s_i = \text{LSTM}([d_{i-1}, \bar{s}_{i-1}, s_{i-1}, \lambda_s]),
\]

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

Table 1: Datasets for NECLMs

<table>
<thead>
<tr>
<th>Data</th>
<th># of words</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>5,209,203</td>
<td>545.41</td>
</tr>
<tr>
<td>Development</td>
<td>517,480</td>
<td>54.21</td>
</tr>
<tr>
<td>Test</td>
<td>64,166</td>
<td>6.42</td>
</tr>
</tbody>
</table>

where \( \lambda \) is the trainable parameter. Then, \( o_i \) is calculated by concatenating the decoder hidden state with a context vector as and a hyperbolic tangent function as

\[
o_i = \tanh([s_i, \bar{s_i}]^\top, \lambda_t),
\]

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

\[
P(w_i|w_{i-1}, s_{i-1}, \bar{s}_i, \Lambda) = \text{SOFTMAX}(o_i, \lambda_o).
\]

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:

\[
\mathcal{L}(\Lambda) = -\sum_{(r', w') \in D} \log P(w'|r'; \Lambda),
\]

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

\[
D = \{(r_1, w_1), (r_2, w_2), \cdots, (r_N, w_N)\}.
\]

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

<table>
<thead>
<tr>
<th>Encoder context hypotheses</th>
<th>training</th>
<th>scoring</th>
<th>%WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>high ASR</td>
<td>single-hyp. high ASR</td>
<td>20.80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>single-hyp. low ASR</td>
<td>21.57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>multi-hyp. average</td>
<td>20.70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>multi-hyp. confidence</td>
<td>20.76</td>
<td></td>
</tr>
<tr>
<td>high WER</td>
<td>single-hyp. high ASR</td>
<td>20.63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>single-hyp. low ASR</td>
<td>21.38</td>
<td></td>
</tr>
<tr>
<td></td>
<td>multi-hyp. average</td>
<td>20.43</td>
<td></td>
</tr>
<tr>
<td></td>
<td>multi-hyp. confidence</td>
<td>20.43</td>
<td></td>
</tr>
<tr>
<td>No rescoring</td>
<td></td>
<td>21.96</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Model</th>
<th>%WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>21.96</td>
</tr>
<tr>
<td>+ NECLM</td>
<td>20.43</td>
</tr>
<tr>
<td>baseline + LSTMLM</td>
<td>20.19</td>
</tr>
<tr>
<td>+ NECLM</td>
<td>19.82</td>
</tr>
<tr>
<td>100-best oracle</td>
<td>14.28</td>
</tr>
</tbody>
</table>

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 enable 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 $K$ in 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 $K = 30-100$.

5.2.2. Evaluation with LSTMLM.

Table 3 shows WERs in different systems un-rescored and rescoring by 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


