Discriminative Bilinear Language Modeling for Broadcast Transcriptions

Akio Kobayashi1, Manon Ichiki1, Takahiro Oku1, Kazuo Onoe1, Shoei Sato1

1NHK (Japan Broadcasting Corporation)
Science and Technology Research Laboratories, Tokyo, Japan
{kobayashi.a-fs, ichiki.m-fq, oku.t-le, onoe.k-ec, satou.s-gu}@nhk.or.jp

Abstract
A discriminative bilinear language model (DBLM) estimated on the basis of Bayes risk minimization is described. The discriminative language model (DLM) is conventionally trained by using n-gram features. However, given a large amount of training data, the DLM is not necessarily trained efficiently because of the increasing number of unique features. In addition, though some of the n-grams share the same word sequences as contexts, the DLM never reflects this kind of information in that they are not designed to work in a coordinated manner. These disadvantages of utilizing n-gram features could lead to a loss of DLM robustness. We solve these issues by introducing a bilinear network structure to the features aimed at factorizing the contexts shared among the n-grams and estimating the model more robustly. In our proposed language modeling, all the model parameters, such as weight matrices, are estimated according to the objective based on the Bayes risk to be minimized on the training lattices. The experimental results show that our DBLM trained in the lightly-supervised manner significantly reduced the word error rate compared with that of the trigram LM, while the conventional DLM does not yield a significant reduction.

1. Introduction
Closed captioning is an expedient access tool for hard-of-hearing people who want to get information from TV. As a public broadcaster, NHK (Japan Broadcasting Corp.) has developed a system for closed-captioning broadcast news using real-time automatic speech recognition (ASR) [1, 2]. In a practical system, all the errors in the ASR results are fixed by operators manually with small delays. Thus, the availability of such applications strongly depends on the ASR performance. Much interest exists in applying discriminative acoustic and language models to achieve higher performance than that of the conventional models. Although, in many studies covering the discriminative language model (DLM) [3–7], the model typically utilizes a set of word/class n-grams (n-tuples) as feature functions, two types of issues arise from the nature of features. One is an explosive increase in the unique number of features for a large amount of training data, and the other is inefficient representations of n-gram features. While some of the n-gram features have common subsequences as contexts (histories), they never cooperate while decoding because the features are not designed to work in a coordinated manner. In addition, information on recognition errors should be reflected not only in the individual words but also in the word contexts. Therefore, the contexts of n-grams should be shared among the features, and they have to be expressed in an efficient manner.

One of the solutions to these issues is taking advantage of the network structure used in the neural network (NN). In the NN-based language model [8, 9], the n-gram context is mapped to a continuous space as an embedding on the hidden layer. Even when massive training data are given, the unique number of contexts can be reduced as long as low-dimensional embeddings are obtained in continuous space. As the model parameters or distinct n-gram features are factorized to be represented efficiently with common subsequences, the model should be estimated more robustly than the conventional one. The network structure is a critical component in designing the DLM, and we apply the structure used in the log-bilinear language model (LBLM) [10, 11]. The LBLM has an NN-like network structure and factorizes the context without the sigmoid function as a non-linear activation in the hidden layer. The capability of this simple network structure was demonstrated by the skip-gram model, which efficiently performs its training even under a massive corpus with large vocabulary [12].

In this paper, we describe a new discriminative language modeling method, which utilizes a simple network structure to estimate a robust model. The structure of the model is similar to that of LBLM, aside from the lack of the softmax function, which typically normalizes network outputs in the LBLM. Instead, by taking advantage of the context factorization, we estimate the parameters of the network (e.g. weight matrices) on the basis of Bayes risk minimization, in order to make the network reflect information about recognition errors. Thus, just like with DLM, the estimated network would naturally be expected to prefer the promising hypotheses in decoding. Our main contribution in this paper is that we demonstrate the efficacy of the discriminative language model employing the network structure. In a set of experiments on transcribing Japanese broadcast programs, the results show that our discriminative language modeling reduced the word error rate compared with that of the conventional DLM. We also compared the results from the models estimated in unsupervised/lightly-supervised manners.

2. Discriminative Language Model
In this section, we give a briefly review of discriminative language modeling using n-gram feature functions [4, 5, 13].

2.1. Overview
A discriminative language model (DLM) is described as a log-linear model [3, 14]. Given an audio input, s, a posterior of
sentence hypothesis, \( w \), is expressed as
\[
P(w|s; \Lambda) = \frac{1}{Z(\Lambda)} \exp \left\{ \lambda_{am} f_{am}(s|w) + \lambda_{lm} f_{lm}(w) + \sum_i \lambda_i f_i(w) \right\},
\]
where \( f_{am}(s|w) \) and \( f_{lm}(w) \) are logarithmic scores given by acoustic/language models, respectively. \( \lambda_{am} \) and \( \lambda_{lm} \) are constant weighting factors. The conventional DLM is given by the last term, \( \sum \lambda_i f_i(w) \), where \( f_i \) denotes a feature function derived from a word n-gram occurring in \( w \), and where \( \lambda_i \in \Lambda \) is a model parameter. The term \( Z(\Lambda) \) denotes a normalization factor.

2.2. Risk Minimization Training on Lattices

2.2.1. Training Objectives

When having training lattices along with references (labeled data), we define an objective based on the Bayes risk [14] for DLM estimation as
\[
L_{\text{risk}}^{(l)}(\Lambda) = \frac{1}{M^{(l)}} \sum_{m=1}^{M^{(l)}} \sum_{w \in L_m} R(w^{\text{ref}}, w) P(w|s_m; \Lambda).
\]
(2)

If no references exist for \( s_m \) (unlabeled data), the aforementioned objective can be generalized in an unsupervised manner [4].
\[
L_{\text{risk}}^{(u)}(\Lambda) = \frac{1}{M^{(u)}} \sum_{m=1}^{M^{(u)}} \sum_{w \in L_m} P(w|s_m; \Lambda) \times \sum_{w' \in L_m} R(w, w') P(w'|s_m; \Lambda),
\]
(3)

where \( R(\cdot, \cdot) \) is an edit distance between two word sequences.

2.2.2. Risk Computation on Training Lattices

The expected risks in Eqs. (2) and (3) are efficiently approximated on the lattices by using edge-wise risks. In the supervised training manner, given a lattice, \( L \), and a reference, \( w^{\text{ref}} \), the edge-wise risk, \( \xi^{(l)}(e) \), at the edge, \( e \), is defined as
\[
\xi^{(l)}(e) = \sum_{e' \in w^{\text{ref}}} o(e, e') \ell_{0,1}(e, e').
\]
(4)

Similarly, the edge-wise risk in the unsupervised manner is defined as
\[
\xi^{(u)}(e) = \sum_{e' \in L} o(e, e') \ell_{0,1}(e, e') p(e'),
\]
(5)

where \( o(\cdot, \cdot) \) is an overlap function between two edges given by
\[
o(e, e') = \min(\tau(e), \tau(e')) - \max(\sigma(e), \sigma(e')), \]
(6)

where \( \tau(e) \) denotes a start node for \( e \), while \( \tau(e) \) represents an end node. \( \ell_{0,1}(\cdot, \cdot) \) is a local cost function defined between overlapping edges and is used in place of the edit distance. In this paper, \( \ell_{0,1}(\cdot, \cdot) \) is a simple binary function defined as
\[
\ell_{0,1}(e, e') = \begin{cases} 0 & \text{if } \text{label}(e) = \text{label}(e') \\ 1 & \text{otherwise} \end{cases}
\]
(7)

In lightly-supervised training, the edge-wise risk is defined as an integration of the risks expressed in Eqs. (4) and (5) so that it relaxes the risk obtained from only the labeled data [13].
\[
\xi^{(p)}(e) = \sum_{e' \in w^{\text{ref}} \setminus e \text{set}} o(e, e') \ell_{0,1}(e, e') + \sum_{e' \in L} o'(e, e') \ell_{0,1}(e, e') p(e'),
\]
(8)

where \( w^{\text{ref}} \) denotes a pseudo reference and \( e \) gives a confidence score of the edge. We used edge posteriors derived from the forward-backward algorithm as confidence scores. The first term is a modified risk of Eq. (4) in that edge-wise risks are computed over the qualified reference edges having higher confidence scores than a threshold \( \alpha \). The second term is derived from Eq. (5), where \( o'(\cdot, \cdot) \) is a modified overlap function, which returns an overlapping frame ratio that excludes frames overlapping with qualified reference edges.

2.2.3. Model Parameter Estimation

Regardless of the risk computation manners, the model parameters, \( \Lambda \), can be obtained by the online iterative algorithm such as stochastic gradient descent (SGD).

For parameter updates, first, the whole risk of the lattice, \( \hat{\gamma} \), is computed using the forward-backward algorithm by accumulating the edge-wise risks, given by any one of Eqs. (4), (5), or (8). Meanwhile, \( \hat{\gamma} \) is decomposed as
\[
\hat{\gamma} = p(e) \gamma(e) + (1 - p(e)) \hat{\gamma}(e),
\]
(9)

where \( \gamma(e) \) is the risk of all the paths passing through \( e \), while \( \hat{\gamma}(e) \) is that of the other paths. The partial derivative w.r.t. \( p(e) \) is given by
\[
\frac{\partial \hat{\gamma}}{\partial p(e)} = \frac{\gamma(e) - \hat{\gamma}}{1 - p(e)}.
\]
(10)

On the other hand, the partial derivative of \( p(e) \) w.r.t. the parameter, \( \lambda_j \), is obtained by
\[
\frac{\partial p(e)}{\partial \lambda_j} = p(e)(1 - p(e)) \psi_j(e).
\]
(11)

\( \psi_j(e) \) is a binary feature function, which returns 1 if \( f_j \) is activated on \( e \). Finally, the partial derivative at \( e \) is given by
\[
\frac{\partial \hat{\gamma}}{\partial \lambda_j} = p(e)(\gamma(e) - \hat{\gamma}) \psi_j(e).
\]
(12)

By accumulating the gradients over the lattice, we can perform the SGD updates for the model parameters.
Table 1: Training Lattices

<table>
<thead>
<tr>
<th></th>
<th>hours</th>
<th>#utts.</th>
<th>#edges</th>
<th>#feats.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>274.2</td>
<td>175.8k</td>
<td>606.7M</td>
<td>99.7M</td>
</tr>
</tbody>
</table>

Table 2: Specs of Training & Evaluation Data

<table>
<thead>
<tr>
<th></th>
<th>hours</th>
<th>#utts.</th>
<th>#words</th>
<th>ppl.</th>
<th>oov</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>2.0</td>
<td>593</td>
<td>22.4k</td>
<td>68.4</td>
<td>0.23</td>
<td>27.3</td>
</tr>
<tr>
<td>Evaluation</td>
<td>1.7</td>
<td>1,319</td>
<td>21.2k</td>
<td>57.6</td>
<td>0.21</td>
<td>37.7</td>
</tr>
</tbody>
</table>

3. Discriminative Bilinear Language Model

3.1. Log-Bilinear Language Model

Prior to a detailed description of DBLM, we briefly review a network structure utilized in the LBLM described in [10, 11]. In typical log-bilinear modeling, n-grams are estimated from a three-layer network, whose inputs are context words (history) and whose outputs are probabilities for predicted words. As depicted in Figure 1, given the vocabulary size, \( V \), the words in the context are represented as \( V \)-dimensional vectors to be fed into the input layer. In the trigram case, the input vectors, \( x^n \) (\( n = 1, 2 \), are projected to the hidden layer by \( H \times V \) weight matrix, \( C^n \), as

\[
y = \sum_n C^n x^n. \tag{13}
\]

\( H \) denotes the number of units in the hidden layer. Note that we omit the bias terms for simplicity. Moreover, the embedding, \( y \), is projected to the output layer by \( V \times H \) matrix, \( D \),

\[
z = Dy. \tag{14}
\]

The output, \( z \), is finally normalized by the softmax function including the hierarchical softmax [11] to get \( n \)-grams as probabilities. Note that the outputs are not normalized in discriminative language modeling in contrast to the LBLM.

3.2. Discriminative Bilinear Language Model

Here, we introduce discriminative bilinear language modeling, which incorporates the network structure of LBLM into DLM. In the track of Eq. (1), the posterior of \( w \) is redefined as

\[
P(w|x; \Theta) = \frac{1}{Z(\Theta)} \exp \left( \lambda_{am} f_{am}(x|w) + \lambda_{lm} f_{lm}(w) \right)
+ \sum_{k=1}^{V} z(h_k) \cdot v(w_k), \tag{15}
\]

where \( \Theta \) is a set of model parameters (weight matrices and biases), \( z(h_k) \) is an output score vector for the history, \( h_k \), and \( v(w_k) \) is a 1-of-\( V \) vector representation of \( w_k \). In contrast with the DLM shown in Eq. (1), the discriminative score of DBLM, the last term of Eq. (15), can be computed from any contexts as long as the predicted words are observed in the training data. In addition, as the embeddings are estimated from the contexts shared by the n-grams, the DLM should be more robust than the DLM. The edge-wise risks are derived in similar ways according to Eqs. (4), (5) and (8) respectively. Analogous to Eq. (11), the partial derivative w.r.t. the element \( d_{ij} \) of \( D \) is given by

\[
\frac{\partial p(e)}{\partial d_{ij}} = p(e)(1 - p(e)) \phi_i(I(e)) y_j. \tag{16}
\]

where \( I(e) \) is an index function which responds to the word associated with \( e \), and \( \phi_i(I(e)) \) is a binary function, which returns 1 if \( i = I(e) \) and 0 otherwise. The final derivative at \( e \) is given by

\[
\frac{\partial \hat{\gamma}_j}{\partial d_{ij}} = \left( p(e)(\gamma(e) - \hat{\gamma}) \phi_i(I(e)) y_j \right). \tag{17}
\]

Taking \( p(e)(\gamma(e) - \hat{\gamma}) \phi_i(I(e)) \) as a propagation error at the output layer, the gradient is equivalent to that derived from the conventional back-propagation algorithm. At hidden layer, setting the error to \( \Delta y_j = \sum_i p(e)(\gamma(e) - \hat{\gamma}) d_{ij} \phi_i(I(e)) \), we can simply update the matrix \( C^n \) by the back-propagation.

Estimating DTLBM is an analogous way of discriminative acoustic modeling in [15] except for risk computation. The efficiency of factorization on the network structure was reported and discussed in [16] for deep neural networks.

4. Experiments

4.1. Setup

NHK’s speech decoder transcribes input audio streams in real time while detecting start and end points of speech segments [17]. The acoustic inputs are parameterized into 39-dimensional vectors: 12 mel-frequency cepstral coefficients (MFCCs) with log-power and their first- and second-order differentials. The decoder uses a two-pass strategy that obtains 1,000-best sentence hypotheses by a bigram LM in the first pass and rescues them using a trigram LM and DLM/DBLM. The acoustic model was trained from Japanese broadcast news, consisting of 600 hours of male/female utterances, followed by minimum phone error (MPE) training. The baseline trigram LM was trained based on Japanese broadcast news manuscripts, transcriptions, and closed captions (620M words), and the vocabulary size was set to 200k.

Table 1 lists the training data taken from broadcast programs for discriminative language modeling. Because of the difficulties in obtaining an overall word error rate (WER) for the training data without exact references, we selected and investigated a 2.0-hour subset from them (Table 2). The number of unique words observed in the training lattices was 145.0k and covered 72.5% of the vocabulary. Discriminative language modeling was conducted on these lattices as training data. Table 2 lists the evaluation data taken from NHK’s broadcast programs (10 episodes) consisting of conversational speech by two cast members. The perplexities, out of the vocabulary rates and the WERs were measured by using the baseline trigram LM.
The discriminative language models were estimated in the unsupervised/lightly-supervised manners. In lightly-supervised training, the biased LMs for generating the pseudo references were constructed from trigram counts of the previously noted baseline corpus and the closed captions (27.3k sentences, 543.9k words). In the experiments, the biased LM was estimated by interpolating two LMs. Decoded by the biased LM, 53.9% of the words in the pseudo references were accepted as “true” when \( \alpha \) was set to 0.9. Table 3 shows the numbers of trigram features utilized by the DLMs when they are varied by using cut-off parameters ranging from 0 to 5. Similarly, Table 4 shows the number of DBLM parameters configured by a variety of hidden-layer sizes. In discriminative modeling, parallelized SGD [18] was performed for integrating individual n-gram features and network parameters. In SGD parameter updating, the momentum factor was set to 0.5, and \( \ell_2 \) regularization was used.

### 4.2. Experimental Results

We compared the results from DLMs and DBLMs under equal conditions, where the number of parameters is virtually the same as the number shown in Tables 3 and 4. Table 5 lists the WERs by the models trained in the unsupervised manner. In this case, the models were estimated on the basis of the risk shown in Eq. (5) without any use of pseudo references. The DBLM (#units=160) achieved the best WER (37.4\%), and it was reduced 0.3% absolutely compared with that of the baseline result. On the contrary, the conventional DLM had about the same WERs (37.5\%) even with the increase in the number of trigram features. Table 6 shows the WER results from the models trained in the lightly-supervised manner. The DBLMs reduced the WERs compared with those of the models trained in the unsupervised manner, while the DLMs did not show any reductions. The DBLMs also proved to reduce word errors better than the DLMs with larger numbers of trigram features. The DBLM (#units=80) reduced 0.2% of WER against the DLM (cutoff=0), while the former model has less number of parameters. It probably indicates that the DBLM obtained the capability of representing word contexts with efficiency to yield the robust discriminative scores. Finally, the DBLM (#units=160) achieved the best WER of 37.3\%, and it reduced 0.4% of the WER absolutely. A matched-pair test [19] showed that the DBLM improved the WERs at a significance level of 0.05 against the baseline results, whereas none of the DLMs achieved the significance level.

### 5. Discussion

We compared two types of discriminative language models in the perspective of the model parameters. Since the conventional DLM utilizes consecutive word sequences (i.e. trigrams) as features unlike the DBLM, the difference between the models would be found in recovered trigrams. Then, we examined the recovery of trigrams for the models trained in the lightly-supervised manner. Table 7 shows the trigram precision rates representing the percentages of correct trigrams for the results of the baseline trigram LM, the DLM, and the DBLM. The recovery and miss rates indicate either the ratios of recovered trigrams or those of missed trigrams against the baseline results. Although the comparison of precision rates apparently shows that the DBLM achieved the best performance for acquiring correct trigrams, the DLM produced a gain of 1.4% in the recovery rate, and it is a higher value than that of the 1.0% obtained by the DBLM. It indicates that the DLM would be more likely to correct trigrams than the DBLM does. However, because the missing rate of the DLM is double that of the DBLM, the DLM tends to miss a larger number of correct trigrams. Such a lack of model robustness would probably be caused by the existence of unsewn trigrams. In the evaluation data, the trigram features of the DLM (cutoff=0) cover 78.0% of the correct trigrams, thus the model never assigns the discriminative scores to the remaining. It would potentially invoke side effects by other trigram features that are activated in the sentence hypotheses obtained in decoding. If the features are activated in the incorrect hypotheses with positive discriminative scores, there is no way of winning for the correct word sequence. Consequently, the DBLM could be more robust than the DLM by reason that it can estimate discriminative scores from any word contexts.

While the DBLM achieved significant gains over the baseline, they remained small against the DLM. One reason is that the discriminative LMs were estimated from the limited amount of training data. Therefore, we will compare the models with a larger amount of training lattices in future work. Additional gains are expected when the DBLM is used together with acoustic models corresponding to the variability frequently observed in conversational speech.

### 6. Conclusion

We designed discriminative bilinear language modeling based on a risk-minimization framework. Our modeling was made to improve ASR performance using a simple three-layer network for estimating a more robust model than the conventional one utilizing n-gram features. Experimental results showed that our discriminative bilinear language model achieved promising results compared with those using conventional n-gram features in lightly-supervised training. In future work, we will examine a more detailed analysis of the network structure and explore the way to achieve efficient discriminative training.
7. References


