Unsupervised Discriminative Language Modeling Using Error Rate Estimator

Takanobu Oba[1], *Atsunori Ogawa*[2], *Takaaki Hori*[2], *Hirokazu Masataki*[1], *Atsushi Nakamura*[2]

[1]NTT Media Intelligence Laboratories, NTT Corporation, Yokosuka, Japan
[2]NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
{oba.takanobu, ogawa.atsunori, hori.t, masataki.hirokazu, nakamura.atsushi}@lab.ntt.co.jp

10.21437/Interspeech.2013-331

Abstract

Discriminative language modeling is a successful approach to improving speech recognition accuracy. However, it requires a large amount of spoken data and manually transcribed reference text for training. This paper proposes an unsupervised training method to overcome this handicap. The key idea is to use an error rate estimator, instead of calculating the true error rate from the reference. In standard supervised approaches, the true error rate is used only for finding the Oracle, the minimum error rate hypothesis, and for prioritizing the competing hypotheses for weighted learning. Namely, we really need the error rate, not the reference. In our proposed method, estimates of the error rate are used instead, and so the references are not necessary. Our experiments show that our proposed method can generate a model that performs to the same level of accuracy as supervised methods.

Index Terms: discriminative language model, unsupervised training, word error rate estimation

1. Introduction

In this decade, discriminative training for language models has received a lot of attention from the automatic speech recognition (ASR) community [1, 2, 3, 4]. Discriminative language models (DLMs) can effectively reduce speech recognition errors, since they are trained from actual recognition results.

Many kinds of learning algorithms have been employed for DLM training. Roark et al. employed the perceptron algorithm and the global conditional log-linear model (GCLM) [1, 5, 6], and Kuo et al. utilized minimum Bayes risk (MBR) training [7]. Zhou et al. compared some learning algorithms such as boosting and ranking SVM [8], and Oba et al. compared weighted learners including minimum error rate training (MERT) using word error rate (WER) as the weight [9]. Miller et al. developed a loss-sensitive algorithm [2], Dikici et al. and Sak et al. proposed ranking perceptron [10] and error sensitive perceptron [11], respectively. Oba et al. proposed a method that distinguishes all sample pairs in a round-robin fashion [3, 12].

DLM training requires both speech data and its reference transcription in the supervised learning framework. First, speech data is recognized with a baseline ASR system to generate competing hypotheses and then a model is trained in a discriminative manner, in which the reference sentences or relatively low WER hypotheses are distinguished from the (other) hypotheses. However, transcribing speech data manually is overly expensive in terms of cost and time.

One way to avoid this weakness is the competitor generation approach, in which *dummy competitors*, i.e. pseudo ASR hypotheses, are generated from a written text. A model is trained assuming the written text as the reference of the dummy competitors. In this approach, it is important to simulate errors that would occur in the actual ASR. Kurata et al. takes into account the phoneme similarities calculated from an acoustic model in generating the competitors [13, 14]. Celebi et al. and Jyothi et al. employed confusion models that were made from actual ASR hypotheses [15, 16]. Sagae et al. used a machine translator where pairs of ASR 1-best output and its reference were regarded as a parallel corpus, and also used directly-extracted competing sub-strings, i.e. cohort, for this purpose [4].

Another approach to eliminate the need to transcribe a large number of speeches manually is semi-supervised learning, in which a few speeches are transcribed manually and the others remain untranscribed. Kobayashi et al. and Tam et al. achieved semi-supervised learning for DLMs, formulating it as a multi-objective optimization programming problem and an adaptation problem between manual transcriptions and recognized transcriptions, respectively [17, 18].

The other approach is unsupervised learning, which only requires unlabeled speech data. To choose better strings (n-grams), Xu et al. relied on a large text corpus, instead of the reference [19]. First, competing n-grams, i.e. a cohort, are obtained from lattices and then their probabilities are estimated in a discriminative manner using the large text corpus. In short, this method assumes that an n-gram that appears many times in the large text corpus would be uttered many times. Kuo et al. investigated an unsupervised approach in the MBR framework [7].

This paper proposes an alternative unsupervised DLM training method. Our proposal relies on the fact that the true reference transcription is used only for calculating the WER of each hypothesis before training. In fact, Roark et al. have revealed the effectiveness of using the Oracle reference, which is the minimum error rate hypothesis, instead of using the true (Gold) reference. The key idea of our proposed method is to use an *error rate estimator*. Estimates of error rate are used for finding the Oracle sentence and for prioritizing competing hypotheses for weighted learning, while the other processes for DLM training are the same as in supervised learning.

Obviously, DLM accuracy depends on the accuracy of the error rate estimator in our proposed method. Therefore, a high accuracy error rate estimator is employed in this paper. In our proposed framework, it is also possible to use the confidence measure of the ASR as an error rate estimator. This is also verified in this paper.

Table 1 summarizes the relationships between the methods described above. The competitor generation approach does not require speech data expect for when making a competitor gen-

Copyright © 2013 ISCA

Table 1: Requirements for supervised, competitor generation, semi-supervised and unsupervised approaches. The competitor generation approach requires written texts and a confusion model. The semi-supervised approach requires transcriptions for a part of speeches.

	Sup.	Comp. gene.*	Semi-sup.	Unsup.*
Text	√	√ (written text)	√ (a part)	
Speech	√		√	√
Other		√ (conf. model)		√

∗Note that both competitor generation and unsupervised approaches are a semi-supervised approach and also a supervised approach in a broad sense. They use speeches and references or a similar source as an 'Other' source, and employ a supervised learning algorithm for parameter estimation.

eration mode, and thus suffers from mismatch between written and spoken languages. Furthermore, obtaining a large number of written texts for a specific task is not easy. These are advantages of our proposed method over the competitor generation approach. In Table 1, the semi-supervised approach is similar to the supervised approach, but it requires a specific learning algorithm, while there are many learning algorithms developed under the supervised approach. In our proposed method, we can choose a learning algorithm that suits the usage environment.

Our experiments show the effectiveness of the proposed method. When employing the error rate estimator proposed by Ogawa et al. [20, 21, 22], our proposed method yielded the same or slightly inferior performance in terms of error reduction rate compared to the supervised approach.

This paper is organized as follows: Section 2 describes DLM with its testing and training in the supervised learning framework. Then, the error rate estimator is detailed in Section 3. Our experiments are described in Section 4 and Section 5 concludes this paper and provides future work.

2. Discriminative Language Models

Let $L = \{h_j | j = 1, 2, \cdots, N\}$ and $\mathbf{f}(h_j)$ be an n-best list from a baseline ASR system and a feature vector obtained from hypothesis h_j, respectively. Specifically, the recognition score (posterior probability in this paper) of h_j is denoted as $f_0(h_j)$.

Assuming discriminative language modeling based on the linear model, the original recognition score and the DLM score are interpolated to decide the final recognition result. This is formulated as follows.

$$h^* = \arg\max_{h \in L} \{a_0 f_0(h) + \mathbf{a}^\top \mathbf{f}(h)\} \quad (1)$$

where $\mathbf{a}$ is a model parameter vector of the DLM and a_0 is a given scaling constant. $^\top$ denotes the inner product operation.

To train a DLM, we have to prepare a data set consisting of:

- N-best lists for I utterances $\{L_i | i = 1, 2, \cdots, I\}$
  Each hypothesis is converted into a feature vector denoted as $\mathbf{f}_{i,j}$. That is, $L_i = \{\mathbf{f}_{i,j} | j = 1, 2, \cdots, N_i\}$.
- References (Oracle sentences)
  We represent the feature vector of a reference sentence as $\mathbf{f}_{i,r}$. However, we use the Oracle, which is the hypothesis sentence with the minimum WER, instead of the true reference [1]. Hence, $\mathbf{f}_{i,r} \in L_i$.
- WER for weighted learning
  The WER of hypothesis $h_{i,j}$ is denoted as $e_{i,j}$.

The final parameter value, $\mathbf{a}$, is decided from the data set, using a learning algorithm. The learning algorithms used here are described below.

Passive-aggressive algorithm 1 (PA1) [23]:
Starting from the zero vector of $\mathbf{a}$, the $\mathbf{a}$ value is updated with every input. Choosing the most likely competitor at a certain time, the $\mathbf{a}$ value is modified so that the competitor is distinguish from the Oracle sentence based on the following equation,

$$\mathbf{a} = \mathbf{a} + \min\left\{C, \frac{1 - \mathbf{a}^\top(\mathbf{f}_{i,r} - \mathbf{f}_i^*)}{||\mathbf{f}_{i,r} - \mathbf{f}_i^*||}\right\}(\mathbf{f}_{i,r} - \mathbf{f}_i^*). \quad (2)$$

$\mathbf{f}_i^*$ is the feature vector of the most likely competitor, which is given by Eq. (1), at each input. PA1 can be regarded as an expansion of perceptron algorithm, in which the update width is constant.

Weighted global conditional log-linear model (WGCLM) [9]:
This method finds the parameter value $\mathbf{a}$ that minimizes the following objective function,

$$\mathcal{O}^{\text{WGCLM}} = \sum_{i=1}^{I} \log \sum_{j=1}^{N_i} \frac{e_{i,j} \exp(\mathbf{a}^\top \mathbf{f}_{i,j})}{\exp(\mathbf{a}^\top \mathbf{f}_{i,r})}. \quad (3)$$

For un-weighted learning, $e_{i,j} = 1$ for all. This objective function is consistent with that of conditional random fields except for the summation with j and weights. The minimization problem can be solved by using a descent gradient method or a quasi-Newton method.

Minimum error rate training (MERT) [24]:
The objective function of MERT forms the expectation value of the error rate, i.e.

$$\mathcal{O}^{\text{MERT}} = \sum_{i=1}^{I} \sum_{j=1}^{N_i} \frac{e_{i,j} \exp(\mathbf{a}^\top \mathbf{f}_{i,j})^\alpha}{\sum_{j'=1}^{N_i} \exp(\mathbf{a}^\top \mathbf{f}_{i,j'})^\alpha}. \quad (4)$$

This function is not concave. To avoid convergence on local minima, the hyperparameter α is employed.

Round-robin duel discrimination (R2D2) [3]:
The objective function is given as

$$\mathcal{O}^{\text{R2D2}} = \sum_{i=1}^{I} \log\left\{ \sum_{j'=1}^{N_i} \sum_{j=1}^{N_i} \frac{\exp(\sigma_1 e_{i,j}) \exp(\mathbf{a}^\top \mathbf{f}_{i,j})}{\exp(\sigma_2 e_{i,j}) \exp(\mathbf{a}^\top \mathbf{f}_{i,j'})} \right\}, \quad (5)$$

where σ_1 and σ_2 are hyperparameters. In the R2D2 framework, the objective function is designed so that all the samples are distinguished in round-robin fashion. To be exact, all the hypothesis pairs in each n-best list are distinguished from each other according to their error rate. In addition, this objective function is concave. Therefore, R2D2 guarantees convergence to the optimum value. The exponential weight, i.e. $\exp(\sigma_2 e_{i,j})$, is introduced to prevent the denominator from becoming zero.

3. Error Rate Estimator

The error rate estimator proposed by Ogawa et al. [20, 21, 22] is employed in this paper. This section summarizes this estimator. While the confidence measure classifies each word into correct or incorrect classes, the error rate estimator undertakes error type classification, in which each word is given one of four labels, (C), (S), (D) and (I), which denote correct recognition, substitution error, deletion error and insertion error, respectively.

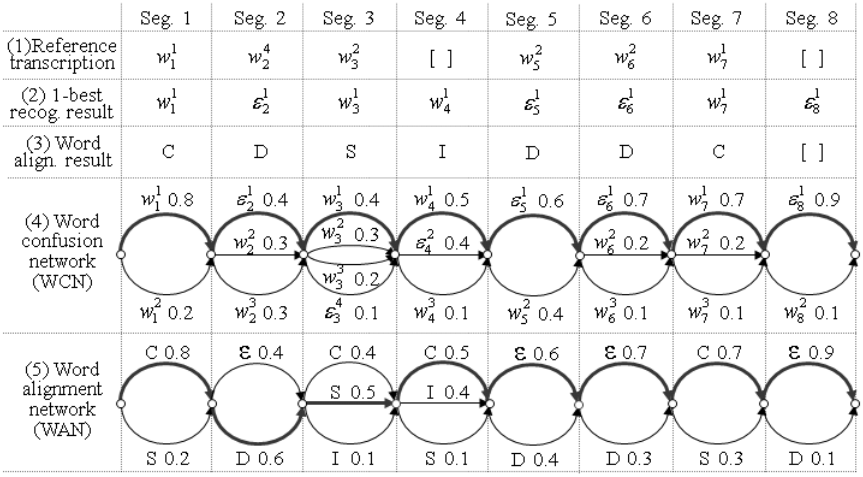

Figure 1: Deriving a *word alignment network* (*WAN*) from a word confusion network (WCN). The target paths are drawn in bold curved lines. The notation w_i^r denotes the r-th rank word at the i-th segment in the WCN.

For labeling, as with the recent trends in confidence estimation, conditional random fields (CRFs) [25] are employed. The final WER is calculated in a probability space as

$$\%\text{WER} = \frac{\sum_{i=1}^{|W|}\{P_i(S) + P_i(D) + P_i(I)\}}{\sum_{i=1}^{|W|}\{P_i(C) + P_i(S) + P_i(D)\}} \times 100, \qquad (6)$$

where $P_i(x)$ denotes the probability of label x at segment (position) i and $|W|$ is the number of words in a given sentence W. Although it is possible to calculate the WER by counting the numbers of the four labels on the best path given by Viterbi decoding, this approach would suffer from labeling errors [20]. To overcome this problem, the WER is calculated in the probability space. Hence, the CRF forward-backward algorithm is required to obtain the label probabilities.

The error rate estimator employs word alignment features (WAFs) as a part of feature vector of CRFs. They are effective for error rate type classification [20]. The WAFs are obtained from a word alignment network (WAN), which is converted from a word confusion network (WCN). The WAN is depicted in Fig. 1. This figure depicts a 1-best recognition result, but it can also be applied to arbitrary (lower-ranking) hypotheses that appear in the WCN [22].

Given the target string and the corresponding WCN, i.e. (2) and (4), the WAN is constructed segment-by-segment. The posterior probability of a target word, which is given from a speech recognizer, is regarded as the probability of the correct recognition label (C), i.e. confidence measure. The remaining posterior probability mass is basically given to substitution label (S). If the null symbol (ε) exists in a segment in the WCN or the target sentence, the posterior probability mass is also distributed among deletion and insertion labels, (D) and (I). These four label probabilities are just WAFs.

In Fig. 1, the target word w_1^1 at segment 1 has the posterior probability of 0.8 in the WCN. Hence, the probability of (C) is 0.8. The remaining posterior probability $1 - 0.8 = 0.2$ is given to label (S), since ε does not appear at segment 1. At segment 2, the target word is ε. In this case, the probability of the label (C) is zero and the probability for label (D) is given by subtracting the occurrence probability of ε from 1.

All the features for the error rate estimator, including the four WAFs, are listed in Table 2. The 18 features are obtained at each segment. These were the same in [22].

Table 2: 18 features in the CRFs. Features 1 to 4 are the four word alignment features.

ID	Feature	ID	Feature
1	Correct recog. prob.	10	Acoustic log like.
2	Substitution error prob.	11	Unigram log like.
3	Insertion error prob.	12	Trigram log like.
4	Deletion error prob.	13	Back-off behavior
5	Recog. word itself	14	# of alternative hyps.
6	Part-of-speech	15	Rank in compet. hyps.
7	Number of frames	16	# of preceding ε segs.
8	Number of phones	17	Sum. of ε probs.
9	# of frames per phone	18	Sum. of # of alt. hyps.

Table 3: Experimental data size. The number of lectures, utterances and words.

	# lects.	#utters.	# words
ERE	207	94,711	1,686,319
DLM	31	19,114	286,840
Dev.	2	2,968	22,166
Eval.	8	6,482	72,283

4. Experiments

We evaluated our proposed unsupervised DLM training method on the MIT lecture speech corpus [26], using the ASR engine, SOLON, which is a weighted finite state transducer based engine [27]. The acoustic model (AM) consisted of standard three-state HMMs with 32-mixture Gaussian pdfs and was discriminatively trained from 104 lectures (110 hours) with a differenced maximum mutual information (dMMI) criterion [28]. The language model (LM) of the SOLON, the modified Kneser-Ney smoothed trigram model, was trained using 150 lectures and some external texts (6,153,006 words in total).

For our experiments, we divided the corpus into 4 sets as described in Table 3. Sets 'ERE' and 'DLM' were used to train

Table 4: WER (%) comparison of proposed method and supervised learning on set 'Eval'. Baseline WER was 27.8%.

	Sup	ERE (proposed)	Cnf
PA1	27.6	27.6	27.8
GCLM	27.2	27.5	27.8
WGCLM	27.3	27.5	27.8
MERT	27.4	27.4	**27.7**
R2D2	**27.0**	**27.2**	27.8

the model of the error rate estimator and DLMs, respectively. No speakers were overlapped between these two sets. The LM of the SOLON was trained with a different set from set 'DLM'. Making the open condition is important to generate an accurate DLM as reported in [1].

To train the model for the error rate estimator, SOLON was first applied to the utterances in set 'ERE' to obtain the WCNs. Then, 5000-best hypotheses were extracted from the WCNs. To reduce the calculation load, a part of them (100 hypotheses approximately) were finally chosen and used for training [22]. These were chosen so that they had different words from each other.

For DLM training, we first generated the WCNs and obtained 5000-best hypotheses from them in the same way as training the error rate estimation model. Next, the true and estimated WERs of each hypothesis sentence were calculated using the true reference and the error rate estimator, respectively. The Oracle sentence was decided in both cases. Hence, the Oracle sentences differ between the supervised and unsupervised approaches. The WERs were also used for weighted learning. After deciding the Oracle sentence, 100 hypotheses were randomly extracted from the 5000-best list as competitors for DLM training. These were consistent between the supervised and unsupervised cases. Choosing competitors randomly provides a better result than using 100-best lists [29].

Counts of word unigrams, bigrams and trigram were used as features. With WGCLM, MERT and R2D2, the L2-norm regularization factor was added to their objective function to avoid overfitting. Moreover, L-BFGS was employed for parameter estimation [30]. The development set was used to tune the hyperparameters, which were the constant C of PA1, the smoothing factor α of MERT, the gain constants σ_1 and σ_2 of R2D2, the L2-norm constant and the scaling factor a_0 for rescoring in Eq. (1). The iteration number of PA1 was fixed to 10 for simplicity. The DLMs were applied to rerank the 5000-best hypotheses with both the development and evaluation sets.

The result is shown in Table 4. The baseline denotes the 1-best result from SOLON. The columns of 'Sup' and 'ERE' show the WERs of the supervised method and our proposed method. For comparison, we evaluated another situation in which the confidence measure (posterior probability) was used in the same way as our proposed method. After calculating the geometric mean of word posterior probabilities in each sentence, the value of $1-$(this is the mean value) was handled as WER, namely, it was used to decide the Oracle sentence and used for weighted learning. This result is shown in the column marked 'Cnf'. If the absolute differences in the WERs on the evaluation set is no smaller than 0.2, they are statistically significant ($p < 0.01$).

The baseline WER 27.8% was decreased by applying the DLM trained using the error rate estimator with all learning algorithms. With PA1 and MERT, their gains were the same as the supervised approach. Since the error reduction rate was significantly degraded with the other algorithms, its gain was slight. The correlation coefficient between the true WERs of n-best hypotheses and those estimated by the error rate estimator is 0.81.

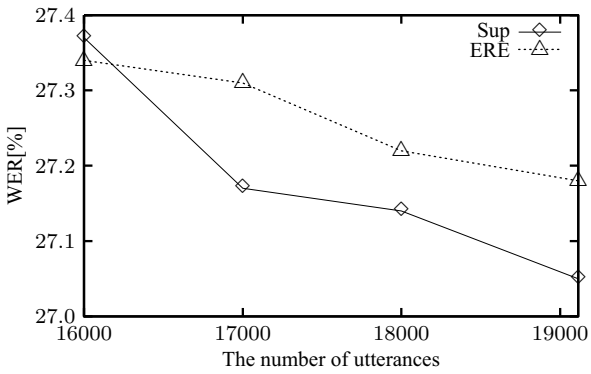

Figure 2: Training data size and WER on the evaluation set.

In contrast, we observed no additional error reduction with the DLMs trained using the confidence measure. The confidence measure is generally regarded as a measure of ASR plausibility. However, the correlation coefficient with the true WER was a mere 0.46. Originally, n-best hypotheses are listed in order of posterior probability, i.e. confidence measure. DLMs were used to rerank them. However, we employed the confidence measure for DLM training. In this situation, keeping the hypothesis order is best for DLM.

Next, the relationship between training data size and WER is depicted in Fig. 2. The data size is given as the number of utterances. The learning algorithm we employed here was R2D2.

The WER was decreased by increasing the training data size. This was consistent with both proposed and supervised methods. As described above, our proposed method has some risk of degrading the WER in comparison with supervised learning. However, we observed that any degradation could be recovered by increasing training data size. In this experiment, that was achieved by adding 1000 to 2000 utterances for training.

5. Conclusion and future work

This paper proposed an unsupervised DLM training method, which is basically the same as the conventional supervised approach except that it estimates the error rate instead of calculating the true error rate. Our experiments showed that our method could basically match the performance of the conventional supervised approach and any slight degradation in accuracy can be offset by increasing the unlabeled utterance data used for training.

In our experiments, the error rate estimation model and DLMs were trained from the same corpus. If the error rate estimation is sensitive to corpora or tasks, a manually transcribed text is required to train the error rate estimation model after all. Furthermore, the data size was much larger for the error rate estimation model than for the DLMs in our experiments. The data size for the error rate estimator has not been investigated yet. These points will be evaluated in future work to confirm the effectiveness of our proposed method.

6. References

[1] B. Roark, M. Saraclar, and M. Collins, "Discriminative n-gram language modeling," *Computer Speech and Language*, vol. 21, no. 2, pp. 373–392, 2007.

[2] N. Singh-Miller and M. Collins, "Trigger-based language modeling using a loss-sensitive perceptron algorithm," in *Proceedings of ICASSP*, 2006, pp. 141–144.

[3] T. Oba, T. Hori, A. Nakamura, and I. Akinori, "Round-robin duel discriminative language models," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 4, pp. 1244–1255, 2012.

[4] K. Sagae, M. Lehr, E. T. Prud'hommeaux, P. Xu, N. Glenn, D. Karakos, S. Khudanpur, B. Roark, M. Saraclar, I. Shafran, D. M. Bikel, C. Callison-Burch, Y. Cao, K. Hall, E. Hasler, P. Koehn, A. Lopez, M. Post, and D. Riley, "Hallucinated n-best lists for discriminative language modeling," in *Proceedings of ICASSP*, 2012, pp. 5001–5004.

[5] B. Roark, M. Saraclar, and M. Collins, "Corrective language modeling for large vocabulary ASR with the perceptron algorithm," in *Proceedings of ICASSP*, 2004, pp. 749–752.

[6] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *Proceedings of ACL*, 2004, pp. 47–54.

[7] H.-K. J. Kuo, E. Arisoy, L. Mangu, and G. Saon, "Minimum Bayes risk discriminative language models for Arabic speech recognition," in *Proceedings of ASRU*, 2011, pp. 208–213.

[8] Z. Zhou, J. Gao, F. K. Soong, and H. Meng, "A comparative study of discriminative methods for reranking LVCSR n-best hypotheses in domain adaptation and generalization," in *Proceedings of ICASSP*, 2006, pp. 141–144.

[9] T. Oba, T. Hori, and A. Nakamura, "A comparative study on methods of weighted language model training for reranking LVCSR n-best hypotheses," in *Proceedings of ICASSP*, 2010, pp. 5126–5129.

[10] E. Dikici, M. Semerci, M. Saraclar, and E. Alpaydin, "Classification and ranking approaches to discriminative language modeling for ASR," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 21, no. 2, pp. 291–300, 2013.

[11] H. Sak, M. Saraclar, and T. Gungor, "Morpholexical and discriminative language models for turkish automatic speech recognition," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 8, pp. 2341–2351, 2012.

[12] T. Oba, T. Hori, A. Ito, and A. Nakamura, "Round-robin duel discriminative language models in one-pass decoding with on-the-fly error correction," in *Proceedings of ICASSP*, 2011, pp. 5588–5591.

[13] G. Kurata, A. Sethy, B. Ramabhadran, A. Rastrow, N. Itoh, and M. Nishimura, "Acoustically discriminative language model training with pseudo-hypothesis," *Speech Communication*, vol. 54, no. 2, pp. 219–228, 2012.

[14] G. Kurata, N. Itoh, and M. Nishimura, "Acoustically discriminative training for language models," in *Proceedings of ICASSP*, 2009, pp. 4717–4720.

[15] A. Celebi, H. Sak, E. Dikici, M. Saraclar, M. Lehr, E. T. Prud'hommeaux, P. Xu, N. Glenn, D. Karakos, S. Khudanpur, B. Roark, K. Sagae, I. Shafran, D. M. Bikel, C. Callison-Burch, Y. Cao, K. Hall, E. Hasler, P. Koehn, A. Lopez, M. Post, and D. Riley, "Semi-supervised discriminative language modeling for Turkish ASR," in *Proceedings of ICASSP*, 2012, pp. 5025–5028.

[16] P. Jyothi and E. Fosler-Lussier, "Discriminative language modeling using simulated ASR errors," in *Proceedings of INTERSPEECH*, 2010, pp. 1049–1052.

[17] A. Kobayashi, T. Oku, T. Imai, and S. Nakagawa, "Multi-objective optimization for semi-supervised discriminative language modeling," in *Proceedings of ICASSP*, 2012, pp. 4997–5000.

[18] Y.-C. Tam and P. Vozila, "A hierarchical Bayesian approach for semi-supervised discriminative language modeling," in *Proceedings of INTERSPEECH*, 2012.

[19] P. Xu, D. Karakos, and S. Khudanpur, "Self-supervised discriminative training of statistical language models," in *Proceedings of ASRU*, 2009, pp. 317–322.

[20] A. Ogawa, T. Hori, and A. Nakamura, "Error type classification and word accuracy estimation using alignment features from word confusion network," in *Proceedings of ICASSP*, 2012, pp. 4925–4928.

[21] ——, "Recognition rate estimation based on word alignment network and discriminative error type classification," in *Proceedings of SLT*, 2012, pp. 113–118.

[22] ——, "Discriminative recognition rate estimation for n-best list and its application to n-best rescoring," in *Proceedings of ICASSP*, 2013.

[23] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[24] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of ACL*, 2003, pp. 160–167.

[25] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of Machine Learning*, 2001, pp. 282–289. [Online]. Available: citeseer.ist.psu.edu/lafferty01conditional.html

[26] J. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT Spoken Lecture Processing Project," in *Proceedings of INTERSPEECH*, 2007, pp. 2553–2556.

[27] 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 Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1352–1365, 2007.

[28] E. McDermott, S. Watanabe, and A. Nakamura, "Margin-space integration of MPE loss via differencing of MMI functionals for generalized error-weighted discriminative training," in *Proceedings of INTERSPEECH*, 2009, pp. 224–227.

[29] T. Oba, T. Hori, and A. Nakamura, "Efficient training of discriminative language models by sample selection," *Speech Communication*, vol. 54, pp. 791–800, 2012.

[30] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 1989.