Pseudo-Conventional N-Gram Representation of the Discriminative N-Gram Model for LVCSR

Zhengyu Zhou, *Member, IEEE*, and Helen Meng, *Senior Member, IEEE*

***Abstract*—The discriminative n-gram modeling approach re-ranks the N-best hypotheses generated during decoding and can effectively improve the performance of large-vocabulary continuous speech recognition (LVCSR). This work recasts the discriminative n-gram model as a pseudo-conventional n-gram model. The recast enables the power of discriminative n-gram modeling to be conveniently incorporated in a single-pass decoding procedure. We also propose an efficient method to apply the pseudo model to rescore the recognition lattices generated during decoding. Experimental results show that when the test data is similar in nature to the training data, applying the pseudo model to rescore the recognition lattices can achieve better performance and efficiency, when compared with discriminative N-best re-ranking (i.e., re-ranking the N-best hypotheses with the discriminative n-gram model). We demonstrate that in this case, applying the pseudo model in decoding can be even more advantageous. However, when the test data is different in nature from the training data, discriminative N-best re-ranking may offer greater benefits than pseudo-model based lattice rescoring or decoding. Based on the pseudo-conventional n-gram representation, we also investigate the feasibility of combining discriminative n-gram modeling with other recognition post-processes and demonstrate that cumulative performance improvements can be achieved.**

***Index Terms*—Discriminative n-gram modeling, large-vocabulary continuous speech recognition (LVCSR).**

I. Introduction

THE use of discriminative training approaches to improve LVCSR performance has received increasing interest in recent years. While state-of-the-art recognizers estimate parameters under the framework of maximum-likelihood estimation, discriminative training approaches adjust the parameters with the aim to directly minimize recognition error rates. Previous approaches include discriminative acoustic modeling [1]–[5], language modeling [6]–[9], or adjustment of the transition weights in the recognition network [10], [11]. In particular, discriminative n-gram modeling [9] has been shown to be effective for both English and Mandarin LVCSR, especially when the test data is similar in nature to the training data [9], [12]. Typically, the N-best hypotheses generated by a baseline recognizer are re-ranked, using a discriminative n-gram model that linearly interpolates the recognition scores with a set of n-gram-based features.

In this paper, we recast the linear discriminative n-gram model as a pseudo-conventional n-gram model. The pseudo model captures the power of the discriminative n-gram model in the sense that integrating the baseline recognizer with the pseudo model generates the same results as the discriminative n-gram model in scoring/ranking utterance hypotheses. The pseudo model can be applied in decoding or lattice rescoring (i.e., rescoring the recognition lattices generated by the baseline recognizer), just like a conventional n-gram model. In this way, the discriminative n-gram modeling can be conveniently extended from distinguishing among the N-best hypotheses to distinguishing among the utterance hypotheses in the decoding search space or in the recognition lattices.

We explore two possible ways to compute the pseudo-conventional n-gram model: offline computation and online computation. The offline computation method builds a complete pseudo model, which is directly applicable in decoding or lattice rescoring. The online method computes pseudo-conventional n-gram likelihoods as needed during application. In this study, we propose an efficient method to compute the pseudo likelihoods online for lattice rescoring. The online computation of the pseudo model for decoding can be similar.

We conduct experiments based on Mandarin dictation. Results show that the effect of discriminative n-gram modeling is sensitive to differences between the training and test sets, which is consistent with previous observations [12]. We demonstrate that when the test data is similar in nature to the training data, applying the pseudo-conventional n-gram model in decoding or lattice rescoring can be advantageous. If the test data is different from the training data, the original approach of using the discriminative n-gram model to re-rank the N-best hypotheses may be more beneficial.

One additional benefit of the pseudo-conventional n-gram representation is that it can conveniently combine discriminative n-gram modeling with other post-processing techniques for recognition. This is illustrated by the use of an error detection and correction framework to post-process the recognition lattices that have been rescored by the pseudo-conventional n-gram model. Our experimental results show cumulative improvements when the training and test data are of similar nature.

The remainder of this paper is organized as follows. Section II briefly reviews discriminative n-gram modeling. Section III presents the pseudo-conventional n-gram model that represents the discriminative n-gram model. An algorithm that efficiently

Manuscript received December 05, 2009; revised December 05, 2009; accepted February 04, 2010. Date of publication August 09, 2010; date of current version November 17, 2010. This work was supported in part by the HKSAR government under Central Allocation CUHK1/02C and also affiliated with the Microsoft-CUHK Joint Laboratory for Human-Centric Computing and Interface Technologies. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Roland Kuhn.

The authors are with the Human-Computer Communications Laboratory, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (e-mail: zhou_zhengyu@ieee.org; hmmeng@se.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2010.2047675

1932-4553/$26.00 © 2010 IEEE

applies the pseudo model to rescore the recognition lattices generated during decoding is proposed in Section IV. Related experiments are discussed in Section V. We investigate the combination of discriminative n-gram modeling with other approaches in Section VI. In Section VII, we present the conclusions and future research directions.

II. Discriminative N-Gram Modeling

A. Linear Framework

Discriminative n-gram modeling defines a linear framework that re-ranks the N-best utterance hypotheses generated by a baseline recognizer [9], [13]. The linear framework can be described as follows.

- The training data set contains m speech utterances and l_i $(i = 1 \ldots m)$ hypotheses for each utterance. Define $x_{i,j}$ as the jth $(j = 1 \ldots l_i)$ hypothesis of the ith utterance. Define $x_{i,R}$ as the hypothesis with lowest character error rate (CER) among $\{x_{i,j}\}$.
- A separate test set of $y_{i,j}$ is defined in a similar way as the training set.
- Define $D+1$ features $f_d(h)$, where $d = 0 \ldots D$ and h is an utterance hypothesis.
- Define a discriminant function as

$$g(h, \vec{a}) = \sum_{i=0}^{D} a_i f_i(h) = \vec{a} \cdot \vec{f}(h). \qquad (1)$$

The task of discriminative training is to find the weight vector $\vec{a}$ that satisfies the following conditions on the test set:

$$g(y_{i,R}, \vec{a}) > g(y_{i,j}, \vec{a}) \quad \forall i \forall j \neq R. \qquad (2)$$

B. Features

For each utterance hypothesis h, the base feature $f_0(h)$ is the recognition score of h. The recognition score is the weighted summation of acoustic and language model (LM) likelihoods that are assigned to the hypothesis in focus by the baseline recognizer. It can be written as follows:

$$f_0(h) = \alpha \sum_{i=1}^{k} P_{AM}(w_i) + \beta \sum_{i=1}^{k} P_{LM}(w_i | w_1, w_2, \ldots w_{i-1}) - k \cdot r \qquad (3)$$

where $w_1 w_2 \ldots w_k$ is the corresponding word sequence of the utterance hypothesis h, $P_{AM}(w_i)$ and $P_{LM}(w_i | w_1, w_2, \ldots, w_{i-1})$ are the acoustic and LM likelihoods (i.e., probabilities in the log domain in this case) for the word w_i, α and β are the acoustic and LM weights adopted by the recognizer, and r refers to the word insertion penalty weighted by the number of words k. The remaining features are the n-gram counts. Given a set of selected n-grams (i.e., n-word sequences), we assign a unique index u $(1 \leq u \leq D)$ to each of the n-grams. For an utterance hypothesis h, $f_u(h)$ is the count of the uth n-gram in h. For example, assuming that the index of the unigram "*new*" is v and that of the bigram "*new solutions*" is w, then for the hypothesis "*There are new ideas and new solutions*," $f_v(h) = 2$ and $f_w(h) = 1$.

```
Initialize the weight vector $\vec{a}$
For $t = 1 \ldots T$ ($T$ is the total number of iterations)
    For the $i^{th}$ speech utterance, $i = 1 \ldots m$
        For $x_{i,j}$ (i.e., the $j^{th}$ hypothesis of the $i^{th}$ utterance), $j = 1 \ldots l_i$
            Calculate $g(h, \vec{a})$, where $h = x_{i,j}$
        Choose the $x_{i,k}$ with the highest $g(h, \vec{a})$ value
        For $d = 0 \ldots D$ ($\eta$ is the size of the learning step)
            $a_d = a_d + \eta(g(x_{i,R}, \vec{a}) - g(x_{i,k}, \vec{a}))(f_d(x_{i,R}) - f_d(x_{i,k}))$
```

Fig. 1. Standard perceptron algorithm with delta rule.

A discriminative M-gram model normally includes all the n-grams with order $n \leq M$ into the calculation of $f_i(h)$ $(1 \leq i \leq D)$. For instance, a discriminative bigram model can utilize both unigrams and bigrams. In addition, since the base feature $f_0(h)$ depends on acoustic likelihoods, discriminative n-gram models are acoustically relevant.

C. Training Algorithm

The weight vector $\vec{a}$ can be trained by different algorithms, including perceptron [14], boosting [15], ranking support vector machine [12], [16], and minimum sample risk [17]. These training algorithms attempt to minimize the training error directly or minimize various loss functions. In this paper, we use the perceptron algorithm as an example of all possible training methods. The perceptron algorithm optimizes a minimum square error (MSE) loss function [18] to approximate the minimum training error. The loss function can be written as

$$f_{\text{loss}}(\vec{a}) = \frac{1}{2} \sum_{i=1 \ldots m} (g(x_{i,R}, \vec{a}) - g(x_{i,k}, \vec{a}))^2 \qquad (4)$$

where $x_{i,k}$ is the utterance hypothesis having the highest $g(h, \vec{a})$ value among all the candidate hypotheses for the ith speech utterance.

In this paper, we follow [17] to use the averaged perceptron algorithm [14], [19] to train the weights. This method first uses the standard perceptron with a delta rule to iteratively update $\vec{a}$, as shown in Fig. 1. The weights are then averaged to increase model robustness. For each weight $a_d (0 \leq d \leq D)$, we define $a_d^{i,t}$ as the value of a_d after processing the ith utterance in the tth iteration. The average weights are calculated as

$$(a_d)_{\text{avg}} = \left(\sum_{t=1}^{T} \sum_{i=1}^{m} a_d^{i,t} \right) / (T \cdot m), \quad d = 0 \ldots D \qquad (5)$$

where T is the total number of iterations.

III. Pseudo-Conventional N-Gram Representation

In this section, we first prove that the linear discriminative n-gram model can be recast as a pseudo-conventional n-gram model [20] in Section III-A. We then discuss the computation of this pseudo-convention n-gram model in Section III-B.

A. Theory

For each speech utterance, the discriminative n-gram model is applied to score/rank the N-best hypotheses using (1). The topranking hypothesis is the new recognition result for the utterance in focus. If a_0 is larger than zero, we can modify the scoring method as (6) without changing the ranking of the N-best hypotheses. Note that given a reasonably good baseline recognizer, a_0 is always positive since $f_0(h)$ (i.e., the recognition score) is a reliable source of information to distinguish among competing N-best hypotheses:

$$g'(h,\vec{a}) = f_0(h) + \sum_{i=1}^{D} \frac{a_i}{a_0} f_i(h). \tag{6}$$

For a discriminative M-gram model that utilizes all the n-grams with order $n \leq M$, the second part of (6) can be expanded into the equation below:

$$\begin{aligned}\sum_{i=1}^{D} \frac{a_i}{a_0} f_i(h) = \frac{1}{a_0} \big(&a_{w_1} + a_{w_2} + \ldots + a_{w_k} + a_{w_1 w_2} \\ &+ a_{w_2 w_3} + \ldots + a_{w_{k-1} w_k} + \cdots \\ &+ a_{w_1 w_2 \ldots w_M} + a_{w_2 w_3 \ldots w_{M+1}} \cdots \\ &+ a_{w_{k-M+1} w_{k-M+2} \ldots w_k}\big)\end{aligned} \tag{7}$$

where $a_{w_p w_{p+1} \ldots w_{p+j}}$ is the weight of the $(j+1)$-gram $(w_p w_{p+1} \ldots w_{p+j})$.

Based on (3) and (7), (6) can be rewritten as

$$\begin{aligned}g'(h) &= f_0(h) + \sum_{i=1}^{D} \frac{a_i}{a_0} f_i(h) \\ &= \alpha \sum_{i=1}^{k} P_{AM}(w_i) \\ &\quad + \beta \sum_{i=1}^{k} P_{LM}(w_i | w_1, w_2, \ldots w_{i-1}) - k \cdot r \\ &\quad + \frac{1}{a_0} \big(a_{w_1} + a_{w_2} + \ldots + a_{w_k} + a_{w_1 w_2} + a_{w_2 w_3} \\ &\qquad + \ldots + a_{w_{k-1} w_k} + \ldots. + a_{w_1 w_2 \ldots w_M} \\ &\qquad + a_{w_2 w_3 \ldots w_{M+1}} \cdots + a_{w_{k-M+1} w_{k-M+2} \ldots w_k}\big) \\ &= \alpha \sum_{i=1}^{k} P_{AM}(w_i) \\ &\quad + \beta \sum_{i=1}^{k} P'_{LM}(w_i | w_1, w_2, \ldots w_{i-1}) - k \cdot r\end{aligned} \tag{8}$$

where

$$\begin{aligned}P'_{LM}(w_i | w_1, \ldots, w_{i-1}) &= P_{LM}(w_i | w_1, \ldots, w_{i-1}) \\ + \frac{1}{a_0 \cdot \beta} &\left(a_{w_i} + a_{w_{i-1} w_i} + \ldots + a_{w_{i-M+1} w_{i-M+2} \ldots w_i}\right).\end{aligned} \tag{9}$$

Comparing (3) and (8), we can see that the only difference in scoring between the baseline recognizer and the discriminative n-gram model lies in the LM likelihoods (i.e., the LM scores assigned to the hypotheses). This means that the discriminative n-gram model can be recast as a pseudo LM (9), and scoring an utterance hypothesis by the discriminative n-gram model is equivalent to scoring the hypothesis by the "upgraded" baseline recognizer that replaces the original LM with the pseudo LM. Hence, the power of the discriminative n-gram model can theoretically be incorporated in a single-pass decoding procedure. While the discriminative n-gram model distinguishes among the N-best hypotheses generated by the baseline recognizer, the upgraded recognizer can directly apply the discriminative knowledge to distinguish among different paths in the decoding search space. Note that the N-best hypotheses are only a subset of the paths in the active decoding search space. Applying the discriminative knowledge to the decoding search space can generate greater benefits, as long as the discriminative knowledge is effective in distinguishing among the N-best hypotheses.

In the discussions above, the baseline LM $P_{LM}(w_i | w_1, \ldots, w_{i-1})$ (i.e., the one in the baseline recognizer) can theoretically be any LM. State-of-the-art recognizers typically adopt conventional n-gram LMs, which estimates the probability of the appearance of a word based on the previous $n-1$ words and can be expressed as $P_{n-gram}(w_i | w_{i-n+1}, \ldots, w_{i-1})$. In the case that the baseline LM is a conventional L-gram model, if $M \leq L$, the discriminative M-gram model can be recast as a pseudo-conventional L-gram model as follows:

$$\begin{aligned}&P'_{L-gram}(w_i | w_{i-L+1}, \ldots, w_{i-1}) \\ &\quad = P_{L-gram}(w_i | w_{i-L+1}, \ldots, w_{i-1}) \\ &\qquad + \frac{1}{a_0 \cdot \beta} \left(a_{w_i} a_{w_{i-1} w_i} + \ldots + a_{w_{i-M+1} w_{i-M+2} \ldots w_i}\right).\end{aligned} \tag{10}$$

Based on (10), the discriminative knowledge can be conveniently incorporated in the decoding procedure. The baseline recognizer can simply use the pseudo-conventional L-gram model instead of the original L-gram model to perform single-pass decoding. In addition, the pseudo-conventional L-gram model can also be applied to rescore the recognition lattices generated by the baseline recognizer, as will be discussed later.

B. Model Computation

To compute the pseudo-conventional n-gram model based on (10), there are two possible methods:

1) Compute the Pseudo-Conventional N-Gram Model Offline: This method attempts to build a pseudo-conventional n-gram model by modifying the n-gram entries in the conventional n-gram model of the baseline recognizer using (10). The advantage of this method is that the application of a precomputed model in both decoding and lattice rescoring is straightforward. For decoding, this model can be applied in the recognizer as a conventional n-gram LM. No change in the structure of the recognizer is needed. For lattice rescoring, standard tools that have been developed for conventional n-gram LMs can be utilized by the pseudo model to rescore the recognition lattices.

The difficulty of this method lies in the fact that some of the n-grams modified by (10) are not included in the baseline conventional n-gram LM. Conventional n-gram LMs do not store all n-grams as entries. This is to make the estimation of n-gram probabilities feasible and to limit memory consumption required in decoding. For an n-gram that is absent but needed in decoding, the baseline n-gram model can compute the probability via back-off to lower-order n-grams, as illustrated in the following example:

$$\begin{aligned}&p(w_i | w_{i-n+1}, \ldots, w_{i-1}) \\ &\qquad = b(w_{i-n+1}, \ldots, w_{i-1}) p(w_i | w_{i-n+2}, \ldots, w_{i-1})\end{aligned} \tag{11}$$

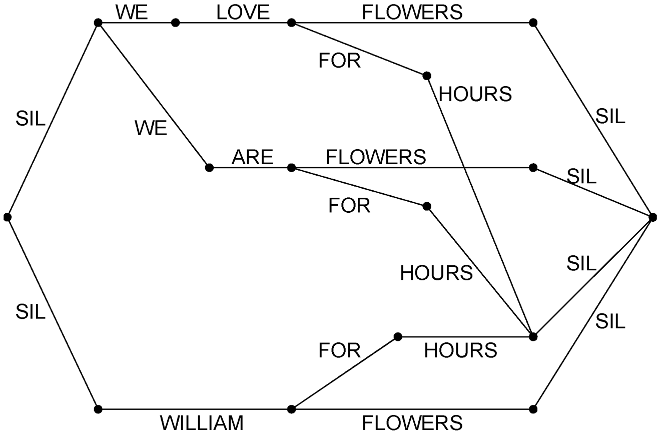


Fig. 2. Sample recognition lattice generated by trigram decoding (SIL marks pauses).

where $p(w_i|w_{i-n+1},\ldots,w_{i-1})$ and $p(w_i|w_{i-n+2},\ldots,w_{i-1})$ are n-gram and $(n-1)$-gram probabilities, respectively. $b(w_{i-n+1},\ldots,w_{i-1})$ is the back-off weight.

One solution to this difficulty is to insert those absent but modified n-grams (i.e., the ones that are not stored in the baseline LM but are modified by (10)) into the pseudo model as new entries. However, the resulting pseudo model may be quite large. Another solution is to adjust the related back-off weights and/or probabilities of lower-order n-grams so that the probabilities of the absent but modified n-grams can be calculated via back-off to lower-order n-grams. In this way, inserting new n-gram entries is unnecessary since they can be computed when needed as in the baseline LM. The model size can thus remain unchanged. We will defer this investigation for the future.

2) Compute the Pseudo-Conventional N-Gram Likelihoods Online: The second method is to compute the pseudo-conventional n-gram likelihoods only when they are needed in either decoding or lattice rescoring. Since no physical model is created in this case, the difficulty encountered in the offline method is circumvented. In this paper, we propose an efficient algorithm to compute the pseudo-conventional n-gram likelihoods online for lattice rescoring, as will be presented in detail in Section IV. For decoding, the online calculation of pseudo-conventional n-gram likelihoods can be similar.

IV. Discriminative Lattice Rescoring

This section presents an algorithm, referred as discriminative lattice rescoring (DLR), to rescore the recognition lattices generated by the baseline recognizer using the pseudo-conventional n-gram model computed online. The procedure for online computation of the pseudo model for decoding is similar since the recognition lattice is a subset of the decoding search space. To facilitate the discussion, we first briefly introduce the properties of recognition lattices. In a recognition lattice, each word hypothesis along with its acoustic and LM likelihoods is stored in a link. If the lattice is generated by a recognizer with L-gram LM, the $(L-1)$-word history for each word hypothesis is unique. Fig. 2 provides a sample recognition lattice generated by trigram decoding.

Suppose the baseline recognizer uses a conventional L-gram LM and a discriminative M-gram $(M \leq L)$ is adopted. The DLR algorithm recasts the discriminative M-gram model into a pseudo-conventional L-gram model as (10) and applies this pseudo model to process the recognition lattices generated by the baseline recognizer as follows:

For each recognition lattice:

Step 1) Traverse all links (i.e., word hypotheses) in the lattice. For each link w_i, compute $(1/(a_0 \cdot \beta))(a_{w_i} + a_{w_{i-1}w_i} + \ldots + a_{w_{i-M+1}w_{i-M+2}\ldots w_i})$ based on the $(M-1)$-word history and add this score to the original LM likelihood (i.e., $P_{L-gram}(w_i|w_{i-L+1},\ldots,w_{i-1})$) stored in this link. The summation is the pseudo-conventional L-gram likelihood $P'_{L-gram}(w_i|w_{i-L+1},\ldots,w_{i-1})$ of w_i.

Step 2) In the rescored lattice, perform the A* search to identify the top-scoring utterance hypothesis. This hypothesis is the new recognition result for the utterance in focus.

In *Step 1*, the $(M-1)$-word history for each link is unique. This is because the $(L-1)$-word history is unique and $M \leq L$. In *Step 2*, the score of an utterance hypothesis $w_1 w_2 \ldots w_k$ in the rescored lattices is

$$Score(h) = \alpha \sum_{i=1}^{k} P_{AM}(w_i) + \beta \sum_{i=1}^{k} P'_{L-gram}(w_i|w_{i-L+1},\ldots,w_{i-1}) - k \cdot r. \quad (12)$$

The utterance hypothesis can be viewed as scored by the upgraded recognizer with a modified L-gram LM. As discussed in Section III-A, this scoring method is equivalent to scoring the utterance hypothesis by the discriminative n-gram model using (1). The top-scoring utterance hypothesis in the rescored lattice is thus the one having the highest $g(h,\vec{a})$ value among all utterance hypotheses in the lattice search space. In other words, performing DLR is functionally equivalent to application of the discriminative n-gram model to re-rank all utterance hypotheses in the recognition lattice, but is more efficient, as will be proven later (see Section V-D).

Note that during the generation of the recognition lattice, some recognizers may merge two word hypotheses that bear the same word and the same time annotations (start and end times) if 1) their LM likelihoods are the same, and 2) the merging will not cause ambiguities in the assignment of L-gram likelihoods for the subsequent word hypotheses. In the recognition lattice generated this way, the $(L-1)$-word history for a link may not be unique. This problem can be solved by various methods. For example, the function of merging word hypotheses satisfying the two conditions above can be disabled for the generation of recognition lattices. Another convenient approach is to insert duplicate links into the recognition lattice to ensure that each link has a unique $(L-1)$-word history.

In the literature, there are many lattice rescoring methods. Most of these methods rescore recognition lattices with models estimated using maximum-likelihood estimation [21], [22].

Compared with such methods, DLR is different in the sense that the adopted model [i.e., $P'_{L-gram}(w_i|w_{i-L+1},\ldots,w_{i-1})$ as (10)] attempts to directly minimize CER, that is, attempts to assign the highest score [i.e., $g'(h)$ as (8)] to the hypothesis with the lowest CER in a lattice.

Another lattice processing approach that aims to minimize word/character error rate is consensus decoding [23], [24]. This approach 1) converts a recognition lattice into a sequence of confusion sets and 2) concatenates the best word (e.g., the word having the highest word posterior probability) of each confusion set to form the output. This technique has been reported to be effective (e.g., reducing the WER from 38.5% to 37.3% on the Switchboard corpus) and widely used in LVCSR systems. While consensus decoding utilizes multiple information sources (e.g., acoustic and LM likelihoods, rules), DLR focuses on LM likelihoods and is relatively convenient to apply.

V. Experiments and Analyses

We conduct experiments on the task of Mandarin dictation. We first train a baseline recognizer that incorporates a conventional trigram LM. We then train a set of discriminative bigram models. Based on these models, we evaluate the original application of discriminative n-gram models (i.e., re-ranking the N-best hypotheses) and the proposed method for discriminative lattice rescoring. We compare the two methods and explore the conditions under which applying the pseudo-conventional n-gram model is suitable.

A. Development of the Baseline Recognizer

We train a general-domain baseline recognizer with state-of-the-art techniques on abundant data. The recognizer utilizes a 60 606-word lexicon for language and acoustic modeling. For language modeling, a conventional trigram LM is trained on a 28-GB text corpus. This text corpus is well balanced across a variety of domains. The LM is smoothed using the absolute discounting algorithm. For acoustic modeling, the trained models are gender-independent cross-word triphone diagonal-covariance Gaussian tied-state HMMs that have 36 Gaussian mixture components [25]. The acoustic models are trained on a 700-hour speech set. In this paper, all the speech datasets involved are read speech recorded in clean environments by Microsoft Research.

B. Development of the Discriminative N-Gram Models

We use a disjoint speech dataset (DT_Set), which contains 84 498 utterances, to train the discriminative n-gram models. In the DT_Set, novels constitute the majority of the content. We refer to this and other similar datasets as the novels-domain datasets in this study.

Since our aim is to investigate the applicability of the pseudo-conventional n-gram model, we did not attempt to develop optimal discriminative n-gram models. Instead, we train two discriminative bigram models based on different numbers of N-best hypotheses, to exemplify different performance levels of the discriminative n-gram models.

The discriminative bigram models utilize the recognition scores, unigram counts and bigram counts as features. Unigrams are the words in the recognizer's lexicon. All the word pairs in the 20-best hypotheses generated by the baseline recognizer for the training set (DT_Set) are used as bigrams. There are 3 657 348 bigrams in total.

The discriminative models are trained using the average perceptron algorithm. a_0 (i.e., the weight for the base feature f_0) is initialized at 0.8, while a_i $(1 \leq i \leq D)$ (i.e., the weight for the feature f_i) is initialized at 0. In the iterative procedure, the size of the learning step is set to be 0.01, and the number of iterations is set at 60. Note that more iterations may lead to better performance especially when the training and test data are similar in nature [12].

As regards the two discriminative bigram models, the first model is trained on the 20-best hypotheses generated by the baseline recognizer for the utterances in DT_Set, while the second model is trained on the 1000-best hypotheses. We refer to the first and second models as Model_20 and Model_1000, respectively. The influence of adopting a greater number of N-best hypotheses in training depends on how well the test data matches the training data [26], as will be discussed later.

The trained discriminative bigram models, Model_20 and Model_1000, can be stored compactly in memory. Although the total number of features (i.e., the recognition score, counts of 60 606 unigrams and 3 657 348 bigrams) is large, the number of active features (i.e., those features whose trained weights are different from the initial weights) is much smaller. Only 12.6% and 17.2% of total features are active for Model_20 and Model_1000, respectively. For each trained model, deleting all inactive features leads to a compact model, which provides the same performance as the original one.

C. Evaluation

We utilize two test sets in evaluation. One is a novels-domain speech set (TestSet_N), which is similar in nature to the discriminative training set DT_Set. The other is a general-domain speech set (TestSet_G) [27], which is different from DT_Set in terms of domain. There are 4000 and 500 utterances in TestSet_N and TestSet_G, respectively. Since discriminative n-gram modeling tends to be sensitive to differences between the training and test data [12], we adopt the two test sets to investigate the applications of the discriminative n-gram models under different conditions (i.e., where the training and test datasets are similar or different in nature).

1) Baseline Performance: For each input speech utterance (e.g., 在新闻中心拜会议长, translation: "Meet the prolocutor at the news center"), we view the recognition result (e.g., 在at/新闻news/中心center/拜会meet/议长prolocutor) as a Chinese character sequence. We then evaluate the recognition performance based on character error rate (CER)

$$CER = \frac{N_S + N_I + N_D}{N_{\text{all}}} \tag{13}$$

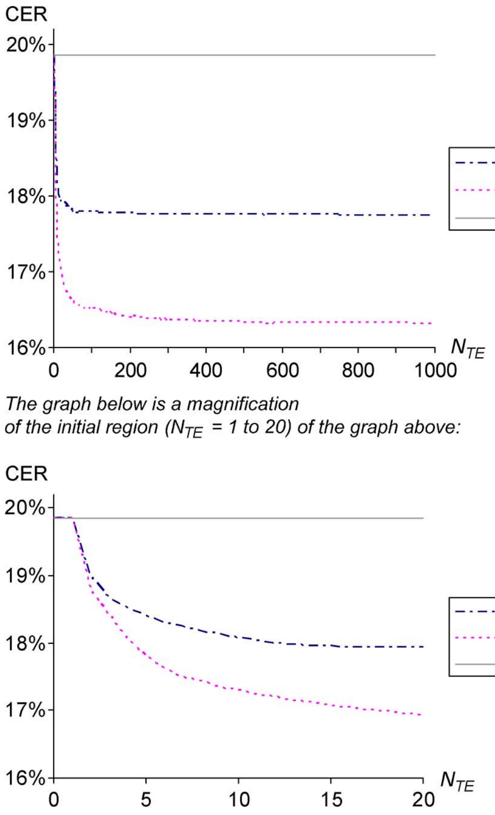
where N_{all} is the number of the characters in the recognized utterances, and N_S, N_I, and N_D are the numbers of character-based substitutions, insertions, and deletions, respectively.

The performances of the baseline recognizer are presented in Table I. We can see that the CER on the novels-domain test set TestSet_N is relatively high. This is because the novels-domain utterances in TestSet_N are dissimilar to those used to train the general-domain LM in the baseline recognizer. The perplexity [28] of the general-domain LM for TestSet_N (novels domain) is more than three times of that for TestSet_G (general domain).

TABLE I
PERFORMANCE OF THE BASELINE RECOGNIZER

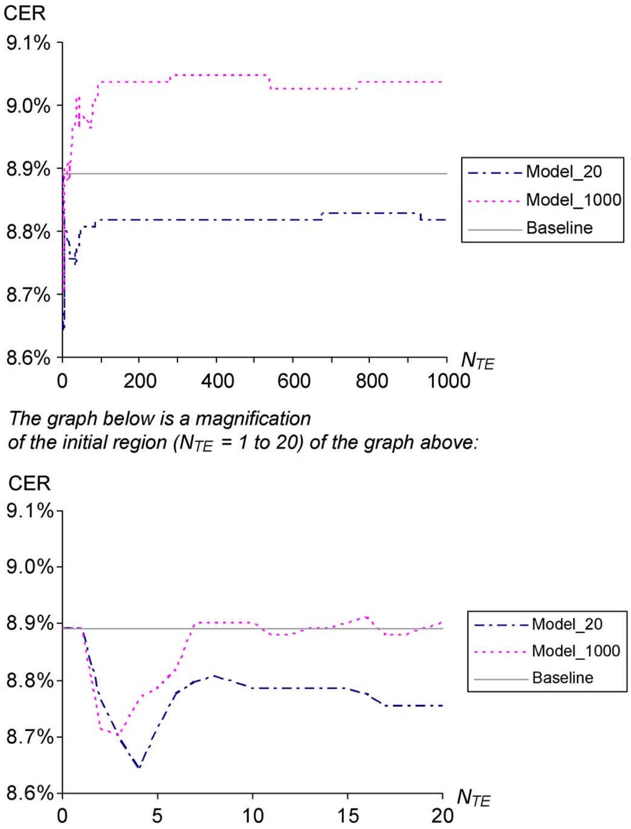
Test Set	Domain	Number of Utterances	Number of Characters	Perplexity	**CER (%)**
TestSet_N	Novels	4,000	62,691	1,528	**19.86**
TestSet_G	General	500	9,572	463	**8.89**

"Perplexity" refers to the perplexity value of the LM used in the recognizer. "CER" is the character error rate [see Equation (13)].



The graph below is a magnification of the initial region (N_{TE} = 1 to 20) of the graph above:

Fig. 3. Performance of discriminative N-best re-ranking on TestSet_N. N_{TE} is the number of the N-best hypotheses for each test utterance.



The graph below is a magnification of the initial region (N_{TE} = 1 to 20) of the graph above:

Fig. 4. Performance of discriminative N-best re-ranking on TestSet_G. N_{TE} is the number of the N-best hypotheses for each test utterance.

2) Discriminative N-Best Re-Ranking: The discriminative n-gram models are used to re-rank the N-best hypotheses generated by the baseline recognizer [see (1)]. We refer to this procedure as discriminative N-best re-ranking. We evaluate the performance of discriminative N-best re-ranking for the two discriminative bigram models, Model_20 and Model_1000 (trained on the 20-best and 1000-best hypotheses, respectively, see Section V-B). We increase the value of N (i.e., the number of N-best hypotheses for each speech utterance in testing) from 1 to 1000. Performance values for TestSet_N and TestSet_G are illustrated in Figs. 3 and 4, respectively.

Fig. 3 shows that for TestSet_N, ranking a greater number of N-best hypotheses for each testing utterance consistently improves the performance for both Model_20 and Model_1000. When N_{TE} increases, the CER drops quickly and then levels off. We can also see that Model_1000 performs better than Model_20 on TestSet_N.

For TestSet_G, Fig. 4 shows that the minimum CERs are achieved by only using the three- or four-best hypotheses in testing. When N_{TE} increases to around 100, the CER curve rises and stabilizes around a relatively high position for both models. N-best re-ranking using Model_1000 mostly hurts performance while Model_20 brings a small improvement only (i.e., less than 0.1% absolute CER reduction).

The different performances between TestSet_G and TestSet_N demonstrate that the effectiveness of discriminative n-gram models depends heavily on how well the test data match the discriminative training data. If the test data (e.g., TestSet_N) is similar to the discriminative training data (e.g., DT_Set), the knowledge captured by the discriminative models is effective in distinguishing among the N-best hypotheses. In this case, adopting a greater number of N-best hypotheses in testing leads to better performance, as observed in the CER reductions in Fig. 3. Furthermore, the greater the number of hypotheses (N) used in training, the more beneficial it will be for testing. Since Model_1000 captures more knowledge than Model_20, it achieves better performance on TestSet_N.

However, if the test data is different from the discriminative training data, the knowledge captured by the discriminative models during training may mislead N-best re-ranking during testing. For example, each discriminative model trained on the novels-domain training data DT_Set captures two types of knowledge: general-domain knowledge and novels-domain knowledge (i.e., knowledge of novels-style expressions, such as "sweet air"). When the model is applied to the general-domain TestSet_G, the general-domain knowledge is beneficial, but the novels-domain knowledge is misleading. More specifically, discriminative N-best re-ranking utilizes both general-domain knowledge (represented as n-gram features trained/tuned on general content) and novels-domain knowledge (represented as n-gram features trained/tuned on novels-style expressions)

TABLE II
PERFORMANCE OF DISCRIMINATIVE LATTICE RESCORING (DLR)

Performance on **TestSet_N** (Novels Domain)		
		CER (%)
Baseline		19.86
Oracle (1000-best hypotheses vs. lattices)		9.44 vs. <8.29
Model_20	**DLR**	**17.74**
	Discriminative 1000-best Re-ranking	17.75
Model_1000	**DLR**	**16.27**
	Discriminative 1000-best Re-ranking	16.31
Performance on **TestSet_G** (General Domain)		
		CER (%)
Baseline		8.89
Oracle (1000-best hypotheses vs. lattices)		3.45 vs. <3.06
Model_20	**DLR**	**8.83**
	Discriminative 1000-best Re-ranking	8.82
Model_1000	**DLR**	**9.06**
	Discriminative 1000-best Re-ranking	9.04

"Oracle" refers to the CER of the best hypotheses (the ones with lowest CERs) in given search spaces (e.g., 1000-best hypotheses or lattices).

that are applied in the N-best hypotheses for TestSet_G. As N increases, the amounts of both types of applied knowledge increase. Fig. 4 indicates that initially, the general-domain knowledge applied grows faster than novels-domain knowledge, leading to minimum CER at around $N_{TE} = 3$ *or* 4. Thereafter, competing effects between the increasing application of general-domain and novels-domain knowledge result in widely fluctuating CER curves. When $N_{TE} > 100$, a rough balance is achieved between the two types of knowledge and the performance curves level off. Furthermore, the fact that Model_1000 performs worse than Model_20 indicates that the former captures a greater amount of misleading knowledge (from the novels-domain) compared with the latter.

The computation time of discriminative N-best re-ranking grows linearly when N increases. 1000-best hypotheses are discriminatively re-ranked in about 0.15 × real-time. In this paper, all computation times were estimated on a server with Pentium 4 CPU of 3.20 GHz.

3) Discriminative Lattice Rescoring: We apply the two discriminative bigram models (i.e., Model_20 and Model_1000) to perform discriminative lattice rescoring. We first recast each model as a pseudo conventional trigram model. We then use the two-step algorithm proposed in Section IV to rescore the recognition lattices generated by the baseline recognizer.

The evaluation results of DLR are illustrated in Table II. We also include the performance of discriminative 1000-best re-ranking in Table II for comparison. Compared with discriminative 1000-best re-ranking, DLR discriminatively re-ranks more hypotheses for each utterance. From Table II, we can see that DLR provides CER reductions similar to discriminative 1000-best re-ranking. The improvements over the baseline brought by both DLR and discriminative 1000-best re-ranking on TestSet_N are statistically significant[1] for both discriminative models. On TestSet_G, the differences with the baseline performance are insignificant for both discriminative models.

Table II also shows that on TestSet_N, DLR slightly outperforms discriminative 1000-best re-ranking. The difference is statistically significant for Model_1000, but is insignificant for Model_20. This indicates that Model_1000 can effectively distinguish among more than 1000 hypotheses when the test data matches the training data. In contrast, Model_20 has almost reached its performance upper bound when the number of hypotheses rises to 1000 (see Fig. 3).

TABLE III
COMPARISON ON TESTSET_G

	Optimal CER (%) of Discriminative *N*-best Re-ranking	CER (%) of DLR
Model_20	8.64 (discriminative *4*-best re-ranking)	8.83
Model_1000	8.70 (discriminative *3*-best re-ranking)	9.06

On TestSet_G, DLR performs slightly worse than discriminative 1000-best re-ranking (see Table II). However, for both discriminative models, the differences between the two approaches are insignificant. As illustrated in Fig. 4, each CER curve fluctuates slightly when $N_{TE} > 100$. Compared with ranking 1000-best hypotheses, the differences brought by DLR (i.e., ranking all hypotheses in the lattices) may be due to chance effects.

Regarding efficiency, the processing speed of DLR is about 0.05 × real-time. For DLR, 88% of the computation time is expended on lattice rescoring and 12% is used for identifying the top-scoring utterance hypothesis in the rescored lattices.

D. Discussion

Our results indicate that it is beneficial to recast the discriminative n-gram model as pseudo-conventional n-gram model and apply the pseudo model in decoding or lattice rescoring if the test data is similar in nature to the discriminative training data. Pseudo-model decoding (i.e., replacing the original LM with the pseudo model in the baseline recognizer to perform decoding) is functionally equivalent to discriminatively ranking the utterance hypotheses in the decoding search space, while DLR is functionally equivalent to discriminatively ranking the utterance hypotheses in the recognition lattice. Note that the N-best hypotheses are subsumed by the recognition lattice and the recognition lattice is subsumed by the decoding search space. Pseudo-model decoding will perform better than DLR and DLR will in turn perform better than discriminative N-best re-ranking (as proven in Section V-C-3) if ranking more hypotheses for each utterance is always beneficial. As shown in Fig. 3, this condition is true when the test data has similar nature as the discriminative training data.

In case that the test data is different in nature from the training data, i.e., when the knowledge captured by the discriminative n-gram model from the training data does not fit the test data well, discriminative N-best re-ranking may perform better than DLR and pseudo-model decoding. This is because incorporating more hypotheses in ranking may introduce a greater amount of misleading knowledge and thus lead to unstable performance (e.g., the fluctuating CER curves shown in Fig. 4). For example, TestSet_G is different from the discriminative training data DT_Set in terms of domain. As demonstrated in Table III, the optimal CERs on this test set are achieved by re-ranking several N-best hypotheses. These optimal CERs are much lower than the CERs of DLR, even though DLR discriminatively ranks all the utterance hypotheses in the recognition lattices. According to the trend of CER when the hypothesis

[1] All the significance tests conducted in this paper are matched-pairs significance tests. The significance level is set at 0.01 throughout this paper.

number grows (see Fig. 4), the performance of pseudo-model decoding should be similar to DLR and also worse than the optimal performance of discriminative N-best re-ranking.

Thus far, three methods have been presented for applying discriminative n-gram models. Discriminative N-best re-ranking and DLR have been implemented. The potential usefulness of pseudo-model decoding has also been discussed. Among the three methods, pseudo-model decoding can be the most efficient if the model is built offline. The recognizer can directly apply the pseudo model in place of the original LM and no additional computation is needed. Online computation of the pseudo-conventional n-gram likelihoods may affect the efficiency of the recognizer. However, DLR with online computation of pseudo likelihoods has been shown to be efficient, costing only 0.05 × real-time to process the lattices. This indicates that the online computation of pseudo likelihoods is sufficiently efficient.

If online computation of pseudo-conventional n-gram models is adopted, we can compare the effectiveness between the pseudo-model based methods (i.e., DLR and pseudo-model decoding) and discriminative N-best re-ranking. When the test data is similar in nature to the training data, methods based on the pseudo model may achieve better performance in less time. For example, DLR performs better than discriminative 1000-best re-ranking on TestSet_G while using only 32% of the computation time needed by discriminative 1000-best re-ranking. When the test data is different in nature from the training data, discriminative N-best re-ranking may be more efficient since the optimal performance may be achieved by re-ranking only a small number of N-best hypotheses for each utterance.

VI. Combination With Recognition Post-Processes

Many recognition post-processes have been proposed to improve LVCSR. Combining discriminative n-gram modeling with other post-processes to achieve cumulative improvements is thus a meaningful topic. Discriminative n-gram modeling is originally a task of N-best re-ranking and it is generally inconvenient to combine with other post-processes (especially for lattice-based processes). However, with the pseudo-conventional n-gram representation, such combination becomes straightforward—other recognition post-processes can be applied to the output of pseudo-model decoding or DLR in the same way as they process the output of conventional decoding. This section provides an illustration by combining an error detection and error correction (ED-EC) framework [26], [29] with DLR.

A. An ED-EC Framework to Improve Mandarin LVCSR

The ED-EC framework attempts to post-process the output of a baseline recognition system by two subsequent procedures: error detection and error correction. Since both procedures depend on recognition lattices, combining the framework with discriminative N-best re-ranking is difficult. However, we can first conduct pseudo-model decoding or DLR and then apply the framework to the discriminatively enhanced baseline systems without any algorithmic modifications. For the discriminatively enhanced baseline systems, the recognition lattices are the ones generated by pseudo-model decoding or rescored by DLR, while the recognized utterances are the top-scoring utterances in these lattices.

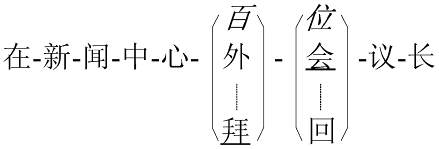


Fig. 5. Search network with character alternatives, created during error correction for an utterance.

In the rest of this section, we briefly introduce the ED-EC framework. Since this framework is only used as an example of combined techniques and is not the focus of this paper, we only present the main ideas of the framework. Detailed information about the framework can be found in [26].

1) Error Detection Procedure: This procedure attempts to detect the erroneous characters in the recognized utterances. First, it labels each Chinese word in the recognized utterances as either correct or erroneous based on the generalized word posterior probabilities (GWPP) calculated from the recognition lattices [26], [30]. Then, for each word that is deemed erroneous, all its characters are labeled as erroneous.

2) Error Correction Procedure: This procedure attempts to rectify the erroneous characters that have been detected, with the help of an advanced LM. For each detected character, a candidate list of character alternatives is created with the aim of including the correct character. The character alternatives are selected from the recognition lattices based on their generated character posterior probabilities [26]. Connecting the candidate lists with the context of the corresponding recognized utterance leads to a new search network, as illustrated in Fig. 5.

An advanced LM is then applied to score the utterance hypotheses contained in the new search network. The advanced LM [26] linearly combines an inter-word mutual information (MI) model, a word trigram model and a POS trigram model, and scores each utterance hypothesis h as follows:

$$S(h) = r_1 \cdot S_{MI}(h) + r_2 \cdot S_{WdTri}(h) + r_3 \cdot S_{PosTri}(h) \quad (14)$$

where $S_{MI}(h)$, $S_{WdTri}(h)$, and $S_{PosTri}(h)$ are the scores assigned to h by the MI model, word trigram model and POS trigram model, respectively, and r_1, r_2, and r_3 are the combination weights.

The candidates in the top-scoring utterance are viewed as the error correction results. For the example of Fig. 5, the two candidates in the top-scoring hypothesis "在新闻中心拜会议长" (Translation: Meet the prolocutor at the news center.) are 拜 and 会.. The corresponding detected errors are corrected.

Note that the error detection procedure may mistakenly label some correct characters as erroneous and "correcting" these *false errors* may introduce new misrecognitions. To address this problem, we adopt an additional mechanism [26] to accept/reject the error correction results based on confidence scores and linguistic scores.

Basically, the ED-EC framework attempts to concentrate the computation of sophisticated LMs on signal segments where a baseline recognition system makes mistakes. If all recognition errors can be perfectly detected, the advanced LM will only be applied to distinguishing among the error alternatives.

TABLE IV
BASELINE RECOGNITION SYSTEMS FOR THE ED-EC FRAMEWORK

Baseline System	Techniques
BL_Orig	Decoding with the original baseline recognizer
BL_DLR$_{20}$	Applying DLR to rescore the recognition lattices using Model_20
BL_DLR$_{1000}$	Applying DLR to rescore the recognition lattices using Model_1000

TABLE V
PERFORMANCE OF THE ED-EC FRAMEWORK OVER DIFFERENT BASELINES

Performance on **TestSet_N** (Novels Domain)			
	Baseline CER (%)	ED-EC Framework CER (%)	Relative CER Reduction (%)
BL_Orig	19.86	19.35	2.6
BL_DLR$_{20}$	17.74	17.36	2.1
BL_DLR$_{1000}$	16.27	16.03	1.5
Performance on **TestSet_G** (General Domain)			
	Baseline CER (%)	ED-EC Framework CER (%)	Relative CER Reduction (%)
BL_Orig	8.89	8.36	6.0
BL_DLR$_{20}$	8.83	8.67	1.8
BL_DLR$_{1000}$	9.06	8.78	3.1

B. Experiments and Analyses

We use a disjoint 2000-utterance speech dataset to develop the error detection procedure and another disjoint 8000-utterance speech dataset to develop the error correction procedure. Both sets are in the domain of novels. For the advanced LM, the three individual LMs (MI, word trigram and POS trigram) are trained on a 340-megabyte general text corpus which consists of the text from the People's Daily and Xinhua newswire in the LDC corpus *the Mandarin Chinese News Text corpus*. The combination weights (i.e., r_1, r_2, and r_3) are tuned by grid search on the development set of error correction.

We apply the ED-EC framework to three baseline recognition systems, as illustrated in Table IV. Two of the systems are related to DLR—the recognition lattices are discriminatively rescored lattices and the recognized utterances are the top-scoring utterance hypotheses in the rescored lattices. For each of the three baseline systems, the ED-EC framework attempts to detect/correct errors in the recognized utterances based on the recognition lattices in the same way.

We develop and evaluate the ED-EC framework for each baseline system separately. The results are shown in Table V. We can see that on the novels-domain TestSet_N, applying the framework effectively reduces the CERs for all the three baseline systems. The improvements over the DLR-related baselines are relatively small because many recognition errors have already been corrected by DLR and the remaining ones are relatively difficult to correct. On TestSet_N, the improvements brought by the ED-EC framework for all the three baseline systems are statistically significant.

On the general-domain TestSet_G, the framework achieves relatively large reductions in CER for BL_Orig. This is because the advanced LM trained on general text is effective in error correction in a general context. The improvement for BL_Orig is statistically significant. However, for the DLR-related baseline systems, the relatively small improvements from the ED-EC framework are statistically insignificant. A possible reason is that for general-domain data, using the discriminative models trained on novels to rescore the recognition lattices inappropriately altered the likelihoods in the lattices, thus affecting the effectiveness of the error detection and correction.

C. Discussion

The previous discussions have demonstrated that it is convenient to combine discriminative n-gram modeling with other recognition post-processes using the pseudo-conventional n-gram representation. In addition to the ED-EC framework, other examples include using an inter-word MI model to re-rank the N-best hypotheses extracted from the discriminative lattices (i.e., the ones generated by pseudo-model decoding or rescored by DLR) [20]. It is also possible to conduct consensus decoding [23], [24] based on the discriminative lattices.

The experimental results reported in the previous subsection indicate that when the test data matches the training data in nature, applying other post-processes to the discriminative lattices may achieve cumulative improvements. However, if the training and test data are different in nature, such benefits may not be observed due to the inappropriately altered likelihoods in the lattices.

VII. CONCLUSION AND FUTURE DIRECTIONS

This work is an extension of discriminative n-gram modeling that applies the discriminative n-gram model to re-rank the N-best hypotheses generated by a baseline recognizer. We prove that a discriminative n-gram model can be recast as a pseudo model, and decoding with the pseudo LM is equivalent to applying the discriminative n-gram model in distinguishing among all utterance hypotheses in the decoding search space. Hence, incorporating the power of the discriminative n-gram modeling into decoding is theoretically possible.

We demonstrate that if the baseline recognizer utilizes a conventional L-gram model, the discriminative M-gram model can be recast as a pseudo-conventional L-gram model if $M \leq L$. Decoding with the pseudo-conventional L-gram model can capture the power of a discriminative M-gram model. The pseudo model can be computed either offline or online. Offline computation faces the difficulty in back-off strategies. We propose an efficient algorithm, called discriminative lattice rescoring (DLR), to compute the pseudo model online to rescore the recognition lattices generated by the baseline recognizer. The pseudo model can be computed in a similar way when it is applied in decoding.

We compare DLR with discriminative N-best re-ranking under different conditions (i.e., the training and test sets are similar or different in nature). Experimental results show that given a discriminative training set in the domain of novels, DLR achieves better performances (i.e., 10.7% and 18.1% relative reductions in CER over the performance of a baseline recognizer) using only 32% of the computation time compared with discriminative 1000-best re-ranking on a novels-domain test set. On the other hand, the optimal performances (i.e., 2.8% and 2.1% relative CER reductions) on a general-domain test set are achieved by discriminatively re-ranking only three- or four-best hypotheses. These demonstrate that it is advantageous to apply DLR or pseudo-model decoding when the test data is similar in nature to the training data. We also investigate the feasibility of applying other recognition post-processes (such as the ED-EC framework) in conjunction with DLR or pseudo-model decoding. Results show that cumulative performance may be achieved.

In the future, we will extend the DLR algorithm to integrate the pseudo-conventional n-gram model online during decoding. We will also investigate offline computation of the pseudo model. A pseudo model built completely offline can be directly applied as a conventional n-gram model for decoding.

ACKNOWLEDGMENT

The authors would like to thank Dr. J. Gao and Dr. F. K. Soong from Microsoft Research for the valuable discussions. They would also like to thank Dr. T. Ye, Dr. C. Huang, Dr. Y. Shi, Dr. F. Seide, Dr. P. Yu, and the late Dr. J. Zhou from Microsoft Research, as well as Dr. W. K. Lo and all other members of the CUHK Human–Computer Communications Laboratory, for their help and support in various ways.

REFERENCES

[1] A. Ben-Yishai and D. Burshtein, "A discriminative training algorithm for hidden markov models," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 3, pp. 204–217, Mar. 2004.

[2] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.

[3] E. McDermott *et al.*, "Discriminative training for large vocabulary speech recognition using minimum classification error," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 203–223, Jan. 2007.

[4] D. Povey *et al.*, "fMPE: Discriminatively trained features for speech recognition," in *Proc. RT'04 Meeting*, 2004.

[5] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 25–47, 2002.

[6] Z. Chen, K. F. Lee, and M. J. Li, "Discriminative training on language model," in *Proc. ICSLP'02*, Denver, CO, 2002.

[7] H.-K. Ku *et al.*, "Discriminative training of language models for speech recognition," in *Proc. ICASSP'02*, Orlando, FL, 2002, vol. I, pp. 325–328.

[8] J. W. Kuo and B. Chen, "Minimum word error based discriminative training of language models," in *Proc. Eurospeech'05*, Lisbon, Portugal, 2005.

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

[10] H. Kuo, G. Zweig, and B. Kingsbury, "Discriminative training of decoding graphs for large vocabulary continuous speech recognition," in *Proc. ICASSP'07*, Honolulu, HI, 2007, vol. IV, pp. 45–48.

[11] S. S. Lin and F. Yvon, "Discriminative training of finite-state decoding graphs," in *Proc. Interspeech'05*, Lissabon, Portugal, Sep. 2005.

[12] 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 *Proc. ICASSP'06*, Toulouse, France, 2006, vol. I, pp. 141–145.

[13] B. Roark, M. Saraclar, and M. Collins, "Corrective language modeling for large vocabulary ASR with the perceptron algorithm," in *Proc. ICASSP'04*, Montreal, QC, Canada, 2004, vol. I, pp. 749–752.

[14] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron Algorithms," in *Proc. EMNLP*, 2002.

[15] M. Collins, "Discriminative reranking for natural language parsing," in *Proc. ICML*, 2000.

[16] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. ACM*, 2002.

[17] J. Gao *et al.*, "Minimum sample risk methods for language modeling," in *Proc. HLT/EMNLP*, 2005.

[18] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[19] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Mach. Learn.*, vol. 37, no. 3, pp. 277–296, 1999.

[20] Z. Zhou and H. Meng, "Recasting the discriminative N-gram model as a pseudo-conventional N-gram model for LVCSR," in *Proc. ICASSP'08*, Las Vegas, NV, Mar. 2008, pp. 4933–4936.

[21] A. Stolcke *et al.*, "Recent innovations in speech-to-text transcription at SRI-ICSI-UW," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1729–1744, Sep. 2006.

[22] S. Matsoukas *et al.*, "Advances in transcription of broadcast news and conversational telephone speech within the combined EARS BBN/LIMSI system," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1541–1556, Sep. 2006.

[23] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Comput. Speech Lang.*, vol. 14, no. 4, pp. 373–400, 2000.

[24] L. Mangu and M. Padmanabhan, "Error corrective mechanisms for speech recognition," in *Proc. ICASSP'01*, Salt Lake City, UT, 2001, vol. I, pp. 29–32.

[25] Y. Tian *et al.*, "Tree-based covariance modeling of Hidden Markov models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 6, pp. 2134–2146, Nov. 2006.

[26] Z. Zhou, "An error detection and correction framework to improve large vocabulary continuous speech recognition" Ph.D. dissertation, Chinese Univ. of Hong Kong, Hong Kong, 2009 [Online]. Available: http://www.se.cuhk.edu.hk/hccl/theses/pdf/2009_ZhouZhengyu.pdf

[27] E. Chang *et al.*, "Speech lab in a box: A mandarin speech toolbox to jumpstart speech related research," in *Proc. Eurospeech'01*, Aalborg, Denmark, 2001.

[28] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit," in *Proc. Eurospeech'97*, Rhodes, Greece, 1997.

[29] Z. Zhou, H. Meng, and W. K. Lo, "A multi-pass error detection and correction framework for Mandarin LVCSR," in *Proc. ICSLP'06*, Pittsburgh, PA, 2006.

[30] W. K. Lo and F. K. Soong, "Generalized posterior probability for minimum error verification of recognized sentences," in *Proc. ICASSP*, Philadelphia, PA, 2005, pp. 85–88.

Zhengyu Zhou (M'07) received the B.S. degree in mathematics from Nanjing University, Nanjing, China, in 1999, the M.S. degree in computer science from Fudan University, Shanghai, China, in 2002, and the Ph.D. degree in systems engineering and engineering management from The Chinese University of Hong Kong (CUHK) in 2009.

She worked as a Project Engineer with the Schlumberger Beijing GeoScience Center in 2003. Her research interest is in large-vocabulary continuous speech recognition, language modeling, discriminative training, spoken document retrieval, and human–computer interface.

Helen Meng (M'98–SM'09) received the S.B., S.M., and Ph.D. degrees, all in electrical engineering, from the Massachusetts Institute of Technology, Cambridge.

She joined The Chinese University of Hong Kong in 1998, where she is currently Professor in the Department of Systems Engineering and Engineering Management and Associate Dean of Research of the Faculty of Engineering. In 1999, she established the Human–Computer Communications Laboratory at CUHK and serves as Director. In 2005, she established the Microsoft-CUHK Joint Laboratory for Human-Centric Computing and Interface Technologies and serves as Co-Director. This laboratory was recognized as one of China's Ministry of Education Key Laboratory in 2008. Her research interest is in the area of human–computer interaction via multimodal and multilingual spoken language systems, as well as translingual speech retrieval technologies.

Prof. Meng serves as Editor-in-Chief for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. She is also an elected board member of the International Speech Communication Association.