RECASTING THE DISCRIMINATIVE N-GRAM MODEL AS A PSEUDO-CONVENTIONAL N-GRAM MODEL FOR LVCSR

Zhengyu Zhou and Helen Meng

The Chinese University of Hong Kong, Hong Kong SAR of China
{zyzhou, hmmeng}@se.cuhk.edu.hk

ABSTRACT

Discriminative n-gram language modeling has been used to re-rank candidate recognition hypotheses for performance improvements in large vocabulary continuous speech recognition (LVCSR). Discriminative n-gram modeling is defined in a linear framework. This work demonstrates that the linear discriminative n-gram model can be recast as a pseudo-conventional n-gram model if the order of the discriminative n-gram model is no higher than the order of the n-gram model in the baseline recognizer. Thus the power of discriminative n-gram model can be captured by mature n-gram related techniques such as single-pass n-gram decoding or lattice rescoring. This work utilizes the pseudo-conventional n-gram model to rescore the recognition lattices that are generated during decoding. Compared to the discriminative *N*-best re-ranking, this process of discriminative lattice rescoring (DLR) has two positive advantages: (1) Those discriminatively top-ranked utterance hypotheses within the lattice search spaces can be efficiently identified by the A* algorithm; (2) The rescored lattices can be further enhanced with other post-processing techniques to achieve cumulative improvement conveniently. Experiments with Mandarin LVCSR show that DLR improves efficiency – the computation time for 1000-best re-ranking is reduced by more than three-fold. The discriminatively rescored lattices are further processed by re-ranking with word-based mutual information (MI). While the DLR achieves around 15% relative character error rate (CER) reductions over the recognizer baseline, the MI based re-ranking further brings 5% relative CER reductions over the DLR performances.

Index Terms—*Discriminative N-gram Modeling, LVCSR*

1. INTRODUCTION

Modern LVCSR systems use the maximum likelihood criterion for parameter estimation. Recently, there is a growing interest in adopting discriminative training methods to enhance LVCSR performance [1,2,3]. While maximum likelihood estimation aims to find the most likely model given the data, discriminative training attempts to minimize recognition error rate. Various discriminative training approaches have been investigated for language modeling. Some methods attempt to adjust the n-gram probabilities [4,5,6]. Other approaches discriminatively model linguistic features to post-process recognition outputs [7,8]. Among these efforts in discriminative language modeling, one algorithm is discriminative n-gram modeling, which selects the counts of n-grams along with the recognition scores as features and defines a global linear model to distinguish among the utterance hypotheses in the *N*-best lists or recognition lattices. This algorithm has been shown to be effective in [7,8,9]. It is also efficient in training, especially compared with discriminative training methods that require iterative decoding [4,6]. Previous work on discriminative n-gram modeling includes the use of weighted finite-state automata (WFA) to store the discriminative n-gram model [8]. By viewing a recognition lattice as an acyclic WFA, finding the discriminatively top-ranked path in the recognition lattice becomes a series of WFA operations including the intersection of two WFAs. Our own previous effort [9] showed that discriminative n-gram modeling can effectively reduce the error rate especially when the training and testing conditions are similar. However, we noticed two bottlenecks: First, the discriminative n-gram model cannot be easily integrated into a single pass decoding procedure. Second, it is not straightforward to extend the discriminative n-gram modeling with other techniques to achieve cumulative improvement. In this work, we recast the discriminative n-gram model into a representation that resembles the conventional n-gram model. Using this pseudo-conventional n-gram model, the power of discriminative n-gram modeling can be conveniently incorporated into a single pass decoding procedure. Such single-pass decoding may deliver a recognized utterance that is not included in the original recognition lattice and/or *N*-best list. This work uses the pseudo-conventional n-gram model to rescore recognition lattices. Within the discriminatively rescored lattices, the best hypothesis (i.e., the utterance hypothesis scored highest by the discriminative n-gram model) can be efficiently identified by A* search. Since the discriminative lattice rescoring procedure outputs lattices, one may conveniently extend it with other post-processing techniques to achieve further improvements.

This paper is organized as follows: Section 2 briefly reviews the discriminative n-gram modeling methodology. Section 3 presents the theories of representing a discriminative n-gram model as a pseudo-conventional n-gram model. Section 4 discusses the application of this pseudo-conventional n-gram model in a new technique known as discriminative lattice rescoring (DLR). Section 5 extends DLR with another post-processing technique that uses word-based mutual information for *N*-best re-ranking. Section 6 describes experiment results and Section 7 gives conclusions.

2. DISCRIMINATIVE N-GRAM MODELING

This section briefly reviews the discriminative n-gram modeling technique, which defines a linear framework to re-rank the *N*-best recognition hypotheses [9]. The modeling process may be described as follows:

- We need a training data set with n speech utterances and n_i utterance hypotheses for each utterance. Define $x_{i,j}$ as the j-th

hypothesis of the *i*-th utterance. Define $x_{i,R}$ as the utterance with lowest CER among $\{ x_{i,j} \}$.

- We need a separate test set of $y_{i,j}$ with similar definitions as the training set.
- Define D+1 features $f_d(h)$, d=0…D, where h is a recognition hypothesis. The features could be arbitrary functions that map h to real values.
- Define a discriminant function as:

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

The task of discriminative training thus involves a search for a weight vector $\vec{a}$ that satisfies the following conditions on the test set:

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

For discriminative n-gram modeling, the features are the recognition scores and the n-gram counts. For each utterance hypothesis h, the base feature $f_0(h)$ is the recognition score which is the weighted summation of acoustic and linguistic likelihoods of h. The remaining features are the counts of each n-gram (i.e., an n-word sequence) in h. We first assign each selected n-gram with a unique id i ($1 \le i \le D$). $f_i(h)$ is then defined as the count of the i^{th} n-gram in h. For instance, the unigram "*new*" and the bigram "*new solutions*" are assigned with ids j and k respectively. Given that h is "*There are new ideas and new solutions*", $f_j(h)$ is 2 and $f_k(h)$ is 1. Normally, a discriminative N-gram model considers all n-grams with order $n \le N$. For example, a discriminative bigram model usually utilizes both unigrams and bigrams.

The weights of the features can be trained by various discriminative training methods [9]. In this work, we utilize the average perceptron algorithm which was described in detail in [10].

3. RECASTING THE DISCRIMINATIVE N-GRAM MODEL AS A PSEUDO-CONVENTIONAL N-GRAM MODEL

For a given speech utterance, the discriminative n-gram model scores each recognition hypothesis as shown in Equation 1 and selects the highest-scoring hypothesis as the recognition result. Unchanging the ranking of hypotheses, we can modify the scoring method as:

$$g'(h,\vec{a}) = f_0(h) + \sum_{i=1}^{D} \frac{a_i}{a_0} f_i(h) \quad (3)$$

For a discriminative N-gram model which considers all n-grams with order $n \le N$, the second part of Equation 3 can be expanded into Equation 4. $w_1 w_2 \ldots w_m$ is the corresponding word sequence of the utterance hypothesis h. $a_{w_i w_{i+1} \ldots w_{i+k}}$ is the weight of the n-gram ($w_i w_{i+1} \ldots w_{i+k}$).

$$\begin{aligned} \sum_{i=1}^{D} \frac{a_i}{a_0} f_i(h) = \frac{1}{a_0} (& a_{w_1} + a_{w_2} + \ldots + a_{w_m} \\ & + a_{w_1 w_2} + a_{w_2 w_3} + \ldots + a_{w_{m-1} w_m} + \ldots . \\ & + a_{w_1 w_2 \ldots w_N} + a_{w_2 w_3 \ldots w_{N+1}} \ldots + a_{w_{m-N+1} w_{m-N+2} \ldots w_m}) \end{aligned} \quad (4)$$

The first part $f_0(h)$ is the score that the recognizer assigned to h, shown as follows:

$$\begin{aligned} f_0(h) &= \sum_{i=1}^{m} (\alpha \cdot AcScore(w_i) + \beta \cdot LmScore(w_i)) - n \cdot InsertPenalty \quad (5) \\ &= \alpha \sum_{i=1}^{m} AcScore(w_i) + \beta \sum_{i=1}^{m} P(w_i \mid w_1, w_2, \ldots w_{i-1}) - n \cdot InsertPenalty \end{aligned}$$

where $P(w_i|w_1, w_2, \ldots, w_{i-1})$ is the log-domain LM likelihood provided by the language model contained in the recognizer. α and β are the acoustic and LM weights for the recognizer respectively.

Combining Equations 4 and 5, Equation 3 can be rewritten as:

$$\begin{aligned} g'(h) &= f_0(h) + \sum_{i=1}^{D} \frac{a_i}{a_0} f_i(h) \quad (6) \\ &= \alpha \sum_{i=1}^{m} AcScore(w_i) + \beta \sum_{i=1}^{m} P'(w_i \mid w_1, w_2, \ldots w_{i-1}) - n \cdot InsertPenalty \end{aligned}$$

where

$$\begin{aligned} P'(w_i \mid w_1, .., w_{i-1}) &= P(w_i \mid w_1, .., w_{i-1}) \\ &+ \frac{1}{a_0 \cdot \beta} (a_{w_i} + a_{w_{i-1} w_i} + \ldots + a_{w_{i-N+1} w_{i-N+2} \ldots w_i}) \end{aligned} \quad (7)$$

Equations 6 and 7 indicate that scoring an utterance hypothesis by the discriminative n-gram model is equivalent to scoring the hypothesis by the recognizer with a modified language model. Suppose the original language model incorporated in the baseline recognizer is a conventional n-gram model with order L. If $N \le L$, we can represent the discriminative N-gram model with a pseudo-conventional L-gram model as shown in the equation below. The power of this discriminative model to distinguish among candidate hypotheses can be captured by the pseudo L-gram model.

$$\begin{aligned} P'(w_i \mid w_{i-L+1}, w_{i-L+2}, .., w_{i-1}) &= P(w_i \mid w_{i-L+1}, w_{i-L+2}, .., w_{i-1}) + \\ & \frac{1}{a_0 \cdot \beta} (a_{w_i} + a_{w_{i-1} w_i} + \ldots + a_{w_{i-N+1} w_{i-N+2} \ldots w_i}) \end{aligned} \quad (8)$$

4. INCORPORATING THE PSEUDO-CONVENTIONAL N-GRAM MODEL IN LVCSR

By recasting the discriminative n-gram model into a pseudo-conventional n-gram model, we can easily incorporate this new language model in direct decoding for recognition or in rescoring recognition lattices.

The pseudo-conventional n-gram model can be computed based on Equation 8 using two possible methods:

(1) Compute the pseudo-conventional n-gram model offline.

A complete pseudo-conventional n-gram model can be built by modifying the n-gram entries in the original n-gram model incorporated in the baseline recognizer using Equation 8. The difficulty lies in the fact that the n-gram model in the recognizer normally does not contain all possible n-grams. This is due to the usage of the back-off strategy for n-gram modeling. Given an n-gram model, an n-gram probability may not be included and be computed via back-off to lower-order n-grams. For example, a bigram not included is calculated as (9). $b(w_1)$ is the back-off weight of w_1.

$$P(w_2 \mid w_1) = b(w_1) P(w_2) \quad (9)$$

For n-grams that are absent from the original n-gram model but are updated by Equation 8, we can insert them into the model as new entries. But this may cause the resulting model to be too large. An alternative method is to keep the model size unchanged and adjust the related back-off weights and/or probabilities of lower-order n-grams. However, the adjustment of backup weights and lower-order n-grams is controversial.

(2) Compute the pseudo-conventional n-gram probabilities online

This approach does not create a physical model and only computes the pseudo-conventional n-gram probabilities when they are

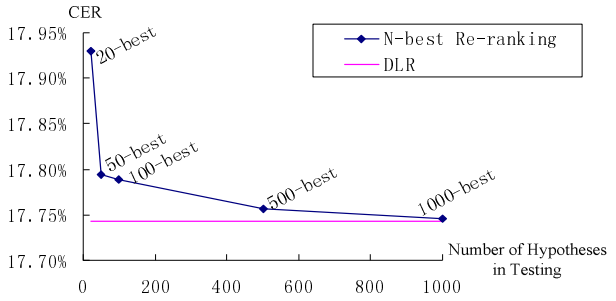

Figure 1. Evaluation using Model_N20

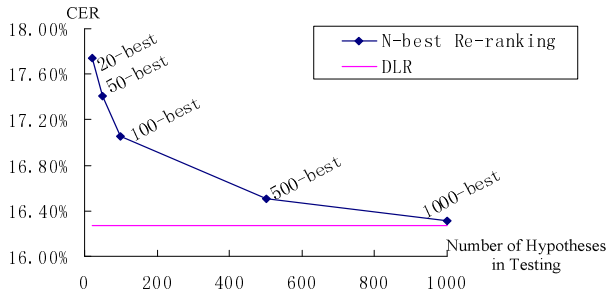

Figure 2. Evaluation using Model_N1000

needed for either decoding or lattice rescoring. Thus, the problem caused by the back-off strategy can be circumvented. Section 4.1 describes the details to compute the pseudo-conventional n-gram probabilities for lattice rescoring. The calculation of pseudo-conventional n-gram probabilities for single-pass decoding is similar.

4.1 Discriminative lattice rescoring

This work uses the pseudo-conventional n-gram model to rescore recognition lattices. We refer this process as discriminative lattice rescoring (DLR). In a recognition lattice, each word hypothesis along with its acoustic and language model (LM) scores is stored in either a link or a node. If the lattice is generated by a conventional *L*-gram model, the (*L-1*)-word history for each word node/link is unique. The basic idea of DLR is to replace the original LM score with the pseudo-conventional n-gram probability for each word node/link in a lattice based on the word history. As shown in Equation 8, the calculation of a pseudo-conventional n-gram probability is composed of two parts: (1) the score from the original n-gram model, and (2) the score from the discriminative n-gram model. Notice that the order of the discriminative n-gram model must be no larger than *L*. For each word node/link, the score from the discriminative model can be calculated easily since the required history is unique. The pseudo-conventional n-gram probability can then be obtained by adding this score to the original n-gram score that has already been stored in the focused node/link.

Having obtained the rescored lattice, the top-scoring utterance hypothesis can be identified efficiently by the A* search. This selected hypothesis is the one having the highest $g(h, \vec{a})$ value among all utterance hypotheses in the lattice search space.

5. EXTENSION WITH OTHER POST-PROCESSING TECHNIQUES

Since DLR outputs a standard lattice representation, it can be extended with other post-processing techniques in a convenient way. In this work, we extend DLR with *N*-best re-ranking based on word mutual information (MI). The MI based re-ranking procedure is applied to each utterance as follows:

1) Select the N-best hypotheses from the corresponding lattice.
2) Assign each hypothesis (i.e., a word sequence $w_1, w_2, \ldots w_m$) with a word mutual information score:

$$MI(w_1, w_2, \ldots w_m) = \left(\sum_{i \neq k} MI(w_i, w_k)\right) / C_m^2 \quad (10)$$

where the mutual information $MI(w_i,w_j)$ is the co-occurrence rate of the two words within an utterance.

3) Score each hypothesis *hypo* by linear interpolation:

$$Score(hypo) = \mu \cdot MI(hypo) + (1-\mu) \cdot LatScore(hypo) \quad (11)$$

where *LatScore(hypo)* is the weighted summation of the acoustic and language model scores extracted from the lattice.

4) The top-scoring hypothesis is the outcome of the re-ranking process.

6. EXPERIMENTS

6.1 Corpora & baseline recognizer

We performed the experiments on Mandarin LVCSR. All speech corpora are in domain of novels. We also utilized a 340 mega-byte LDC corpus "Mandarin Chinese News Text corpus", referred as LM_data, to train the word mutual information mentioned in Section 5. Table 1 shows the information about the training, development, and testing sets in this study.

	Name	Utterances
Training Set	Tr_Set	84,498
Development Set	Dev_Set	2,000
Test Set	Test_Set	4,000

Table 1. Data sets

The baseline LVCSR is a state-of-the-art decoder. The cross-word triphone acoustic models were trained on a separate Mandarin dictation speech corpus of about 700 hours, collected by considering the distribution of gender and age throughout the recording. A trigram model was trained on about 28G (disk size) domain-balanced text corpora, using a 60606-word lexicon. This baseline decoder provides a 19.86% character error rate (CER) on Test_Set. We measure CER by the edit distance of character in this study.

6.2 Development of the discriminative n-gram models

We adopt discriminative bigram modeling in this study. The features include the recognizer score and the counts of unigrams and bigrams. Since it was shown in [7,8] that the benefit of adding trigram features is limited, we focus on unigrams and bigrams for simplicity and efficiency. We used the lexicon entries in forming unigrams. All the word pairs in the 20-best hypotheses of the training data Tr_Set were included as bigrams.

With these features, we trained discriminative models on Tr_Set using the average perceptron algorithm. We initialized the weight for the base feature (i.e., the recognition score) at 0.8. The weights for other features were initialized as 0. All the feature weights were updated in the following way during the training procedure: We set the size of the learning step to be 0.01 and the iteration number at 60. More iterations may lead to better performance, but we did not optimize the iterations in this study.

We varied the number of *N*-best hypotheses based on the training set and evaluated the resulting models on the test data. Results show that performance increases with *N*, the number of training hypotheses used. We selected the model trained on 20-best hypotheses, named Model_N20, as well as the model trained on 1000-best hypotheses, named Model_N1000, to cover discriminative models with different levels of effectiveness. Following experiments are focused on the two models.

6.3 Evaluation: *N*-best re-ranking vs. DLR

The original application of a discriminative n-gram model is re-ranking the *N*-best hypotheses, referred as discriminative *N*-best re-ranking. We compared the discriminative lattice rescoring (DLR) approach with the discriminative *N*-best re-ranking method on the Test_Set. For either Model_N20 or Model_N1000, we first directly used the focused discriminative model to re-rank various numbers of testing *N*-best hypotheses. Then, we recast the focused model as a pseudo-conventional trigram model and performed discriminative lattice rescoring. Results for Model_N20 and Model_N1000 are shown in Figure 1 and Figure 2 respectively. For discriminative *N*-best re-ranking, re-ranking more hypotheses brings better performance for either Model_N20 or Model_N1000, partially because the training and testing conditions are similar. Performing DLR is functionally equivalent to re-ranking all hypotheses in the lattice space, but is more efficient. For DLR, the discriminatively top-ranked hypothesis in a lattice is identified within 0.25s on average. As a reference, discriminative 1000-best re-ranking that provides slightly worse performance than DLR takes 0.78s on average to process a speech utterance. Re-ranking all hypotheses to find the best hypothesis is even more time-consuming. In this study, all computing times are obtained with a Pentium 4 CPU of 3.20GHz. For both the DLR and *N*-best re-ranking, most of the computation is devoted to calculating discriminative scores. The computational load is thus mainly determined by the number of word hypotheses in focus. There are on average 2,908 word hypotheses (nodes/links) in a lattice and 12,120 word hypotheses in the *N*-best (*N*=1000) hypothesis lists of an utterance.

6.4 Extending DLR with the word MI re-ranking

We extended DLR with a further re-ranking procedure based on word MI in an attempt to achieve cumulative improvement. We trained a word MI model on the LM_data using the 60606-word lexicon. We then applied the MI based 100-best re-ranking on the baseline from the original decoder as well as on the enhanced baselines from the DLR lattices rescored with Model_N20 and Model_N1000. For each baseline, the interpolation weight μ was tuned on the Dev_Set. The re-ranking results on the Test_Set are shown in Table 2. We observed several percentage points of relative improvement across various performance baselines, indicating that technique combination based on discriminative lattice rescoring is feasible.

		MI Re-ranking %		Relative Reduction%
		Before	After	
Decoder Baseline		19.9	18.5	7.0
DLR	Model_N20	17.7	16.8	5.1
	Model_N1000	16.3	15.5	5.0

Table 2. CERs on various baselines

7. CONCLUSIONS AND FUTURE RESEARCH

This work extends the discriminative n-gram modeling technique by recasting the linear discriminative n-gram model in a pseudo-conventional n-gram model. This recast model enables easy incorporation in the decoding process of speech recognition. Alternatively, the recast model can also be applied to rescore recognition lattices generated during decoding. The best hypothesis (i.e., the hypothesis scored highest by the discriminative n-gram model) in the lattice space can thus be identified efficiently by A* search. Based on the rescored lattices, the processing can also be extended further with additional lattice post-process.

In this work, we use the pseudo-conventional n-gram model to rescore recognition lattices. We refer this process as discriminative lattice rescoring (DLR). Experiments with Mandarin LVCSR show that DLR can identify efficiently the best hypothesis in lattice, when compared to discriminative *N*-best re-ranking. We extended the DLR processing further with re-ranking by word mutual information and achieved cumulative improvements in recognition performance. In our future research, we will integrate the pseudo-conventional n-gram model into a single-pass decoding procedure.

8. ACKNOWLEDGMENTS

We thank Dr. Jianfeng Gao from Microsoft Research Asia for the valuable suggestions. We also thank Dr. T. Ye, Dr. Y. Shi, Dr. F. Seide, Dr. P. Yu and Dr. F. K. Soong from MSRA for their various help and support. This project was partially supported 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.

9. REFERENCES

[1] P. C. Woodland and D Povey, "Large Scale Discriminative Training for Speech Recognition," *ASR-Speech Recognition: Challenges for the Millenium*, pp. 7-16, Paris, 2000.

[2] A. Ben-Yishai and D. Burshtein, "A Discriminative Training Algorithm for Hidden Markov Models," *IEEE Transactions on Speech and Audio Processing*, vol12(3), pp.204-217, 2004.

[3] S. S. Lin and F. Yvon, "Discriminative training of finite-state decoding graphs," *Proc. InterSpeech*, 2005.

[4] Z. Chen, K. F. Lee and M. J. Li, "Discriminative Training on Language Model," *Proc. ICSLP*, 2002.

[5] H. K. J. Kuo, E. Fosler-Lussier, H. Jiang, C. H. Lee, "Discriminative Training of Language Models for Speech Recognition," *Proc. ICASSP,* 2002.

[6] J. W. Kuo and B. Chen, "Minimum Word Error Based Discriminative Training of Language Models," *Proc.* Eurospeech, 2005.

[7] B. Roark, M. Saraclar and M. Collins, "Corrective Language Modeling for Large Vocabulary ASR with the Perceptron Algorithm," Proc. *ICASSP*, 2004.

[8] B. Roark, M. Saraclar and M. Collins, "Discriminative n-gram language modeling," *Computer Speech and Language,* 21(2):373-392, 2007.

[9] 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," Proc. *ICASSP*, 2006.

[10] J. Gao, H. Yu, W. Yuan and P. Xu, "Minimum sample risk methods for language modeling," Proc. *HLT/EMNLP,* 2005.