Constrained Discriminative Training of N-gram Language Models

Ariya Rastrow [#1], Abhinav Sethy [*2], Bhuvana Ramabhadran [*3]

[#] *Human Language Technology Center of Excellence, and*
Center for Language and Speech Processing, Johns Hopkins University, MD, USA
[1] ariya@jhu.edu

[*] *IBM T.J. Watson Research Center, Yorktown Heights, NY, USA*
[2] asethy@us.ibm.com
[3] bhuvana@us.ibm.com

***Abstract*—In this paper, we present a novel version of discriminative training for N-gram language models. Language models impose language specific constraints on the acoustic hypothesis and are crucial in discriminating between competing acoustic hypotheses. As reported in the literature, discriminative training of acoustic models has yielded significant improvements in the performance of a speech recognition system, however, discriminative training for N-gram language models (LMs) has not yielded the same impact. In this paper, we present three techniques to improve the discriminative training of LMs, namely updating the back-off probability of unseen events, normalization of the N-gram updates to ensure a probability distribution and a relative-entropy based global constraint on the N-gram probability updates. We also present a framework for discriminative adaptation of LMs to a new domain and compare it to existing linear interpolation methods. Results are reported on the Broadcast News and the MIT lecture corpora. A modest improvement of 0.2% absolute (on Broadcast News) and 0.3% absolute (on MIT lectures) was observed with discriminatively trained LMs over state-of-the-art systems.**

I. Introduction

In many natural language processing (NLP) systems, such as Automatic Speech Recognition (ASR) and Machine Translation (MT), a language model is the crucial component for identifying the correct hypothesis in the often prohibitively large hypothesis space. Statistical Language Models (SLMs) are typically trained using Maximum likelihood estimation on vast quantities of text that represent the domain of interest [1].

Recently, discriminative language modeling has been the focus of research [2], [3], [4]. It is natural to expect that SLMs can benefit from discriminative training given their role in selection of the correct hypothesis from the output hypothesis space of NLP systems. These methods attempt to capture the acoustic confusion in the decoded hypotheses by minimizing the training recognition error in constructing the SLMs. It can be seen from [5] that such LMs are useful for Out-of-Vocabulary (OOV) detection task, in addition to improving the overall performance of a speech recognition system.

A model-based approach for discriminative training of N-gram language models, based on minimizing a misclassification function which captures the difference of the likelihood between the reference path and the N-best hypotheses, is proposed in [3]. The following questions arise when using this method:

- **Updating Back-off N-grams** for which there is no explicit model in the initial LM,
- **Normalization** to ensure that the updated N-grams conform to a valid probability distribution, and
- **Unconstrained Updates** that allow local N-gram updates with no global constraints.

In this paper, we specifically address the above-mentioned issues using a two-step procedure described in Section III.

The rest of the paper is organized as follows: Section II recaps the general frame work of the method proposed in [3] for obtaining the N-gram updates. Section III introduces the proposed enhancements to discriminatively updating the language model. Section IV presents our method as a discriminative adaptation framework. Section V describes the experimental setup and the results are analyzed in Section VI. Section VII summarizes the ideas discussed in this paper.

II. Discriminative updates for N-grams

In this section, we present the algorithm for computing updates for N-gram probabilities in a discriminative fashion. Consider a N-gram LM built using techniques described in [1]. An Automated Speech Recognition (ASR) system is used to decode the training data and generate multiple hypothesis that will subsequently be compared to the reference transcripts. The N-gram updates are obtained by comparing the correct word sequence and the corresponding N-best list generated by the ASR system. The technique developed in [3] is based on optimizing a misclassification function that is used to quantify the error rate of an utterance. For a given observation X_i representing the speech signal and a word sequence $\hat{W} = w_1, w_2, \cdots, w_n$ the discriminant score used during decoding is a weighted combination of acoustic and language model scores:

$$g(X_i, W; \Lambda, \Gamma) = \alpha \log P(X_i|W, \Lambda) + \log P(W|\Gamma) \ . \quad (1)$$

Here Λ is acoustic model, Γ is the language model and α

is the inverse of the language model weight. The misclassification function is defined as follows:

$$d(X_i;\Lambda,\Gamma) = -g(X_i,W_0;\Lambda,\Gamma) + G(X_i,W_1,\cdots,W_N;\Lambda,\Gamma)\ , \quad (2)$$

Here $W_1,\cdots,W_N$ corresponds to the N-best list hypotheses and W_0 is the reference word sequence. The anti-discriminant function based on the N-best list competitors is defined as:

$$G(X_i,W_1,\cdots,W_N;\Lambda,\Gamma) = \log(\frac{1}{N}\sum_{r=1}^{N}\exp[g(X_i,W_r;\Lambda,\Gamma)\eta])^{\frac{1}{\eta}}\ .$$

where, η controls the weighting of the different hypotheses in the N-best list (In the limit, as $\eta \rightarrow \infty$ the anti-discriminant function is dominated by the score of the top hypothesis in the lattice). An error in the recognition of the utterance renders $d(X_i;\Lambda,\Gamma) > 0$ i.e. the discriminant function for the correct word sequence scores less than the anti-discriminant function of its competing word sequences. A Generalized probabilistic descent (GPD) [6] based algorithm is used to determine the N-gram updates and adjust the parameters of the language model.

III. Algorithm for Applying N-gram Updates to the Language Model

The procedure described in the previous section provides the N-grams and their corresponding updates (in log probabilities) which need to be incorporated into the initial LM. This section describes the proposed method while specifically addressing the important issues of normalizing and constraining the updates that were introduced in Section I.

Figure 1 illustrates the steps in the proposed method. First, the updates for N-grams are generated using the method described in the previous section. All updates are backed-off to the existing N-gram events of the LM. This is described more in Section III-A. Then the loop for applying the updates to the initial LM starts. The loop includes steps for *Normalizing updated probabilities* and *Relative Entropy Constraint*. These steps are described in Sections III-B and III-C, respectively.

A. Updating Missing N-grams

Quite often there are updates for the N-grams for which there are no explicit models in the initial LM and the probability of the N-gram has to be calculated using the back-off model. Let $S(h)$ (h is the history of the N-gram) denote the set of words for which the probability estimate $p(w|h)$ is explicitly defined in the initial LM. Consider the N-gram probabilities $p(w|h)$ which lie in the complement set denoted by $S^c(h)$. The probability of these N-grams is represented using the probability of the back-off N-grams with history $h' = w_2..w_{n-1}$ as,

$$p(w|h) = \beta(h) * p(w|h') \quad (3)$$

where $\beta(h)$ is the back-off weight.

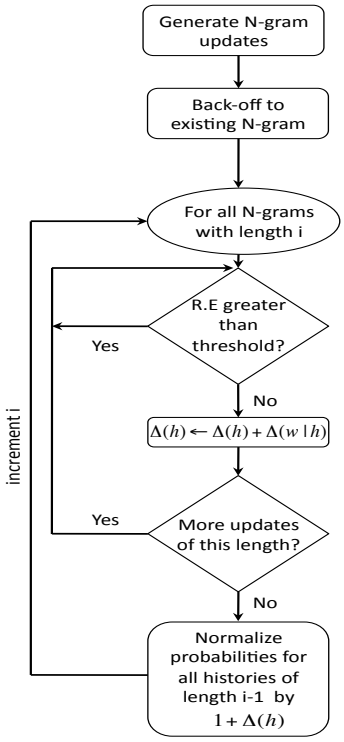

Fig. 1. Algorithm for applying updates to the language model

Two methods for updating these N-grams can be considered:

- In the first method, the N-gram is added to the LM using it's back-off probability and subsequently updated. For example, $p(w|h)$, is computed using Equation 3, and added the LM. This probability is then updated using the update for the N-gram.
- In the second method, the back-off probability which is used to obtain the probability of the new N-gram is updated. For example, the update is applied to $p(w|h')$ if there is an explicit model for $p(w|h')$. Otherwise, the method recursively backs-off until an explicit model is found.

In this paper, the second method is chosen for its ease of implementation and does not result in an increase in the LM size with the addition of new N-grams. Updating the back-off probabilities affects a larger number of N-grams and could have a bigger impact on the performance. Therefore, there is a need for constraining the updates. These constraints can be imposed by algorithms that cluster N-grams into decision-trees or topics or with the use of global constraints such as the one described in this paper.

B. Normalization

The optimization procedure described in Section II does not impose any constraint on the updates such that the updated LM stays a probability distribution, i.e.,

$$\sum_w p(w|h) = 1$$

To guarantee this, the N-gram probabilities need to be normalized after applying the discriminately determined updates. However, the fact that the normalization needs to be done for all possible histories h, for which there was at least one update as (w, h), makes it computationally expensive. A faster method is proposed below. For a given history h, Equation 3 can be rewritten to satisfy a probability distribution as:

$$\sum_{w \in S(h)} p(w|h) + \beta(h) \left(1 - \sum_{w \in S(h)} p(w|h') \right) = 1 \qquad (4)$$

Subsequent to the updates for all $p(w|h)$ Equation 4 simplifies to:

$$\begin{aligned} &\sum_{w \in S(h)} p'(w|h) + \beta(h) \left(1 - \sum_{w \in S(h)} p(w|h') \right) = \\ &\sum_{w \in S(h)} p(w|h) + \sum_{w \in S(h)} \Delta(w|h) + \\ &\beta(h) \left(1 - \sum_{w \in S(h)} p(w|h') \right) = 1 + \sum_{w \in S(h)} \Delta(w|h) \end{aligned} \qquad (5)$$

where,

$$p'(w|h) = \begin{cases} p(w|h) & (w,h) \quad \text{if not updated} \\ p(w|h) + \Delta(w|h) & (w,h) \quad \text{if updated} \end{cases}$$

Therefore, the normalization factor for all explicit probabilities $p(w|h)$ and back-off weights is given by $1 + \sum_{w \in S(h)} \Delta(w|h)$, where $\Delta(w|h)$ is the update for w given the history h. This makes it computationally feasible as only the total set of updates needs to be stored.

Equation 4 is valid only if $p(w|h')$ is already normalized and is a valid probability distribution. This suggests that the procedure of updating N-grams should begin with the lower order N-grams and expand to the higher order N-grams.

C. Relative Entropy based Global Constraint

One of the main problems with the technique described in Sections II is the fact that N-gram features (updates) are obtained based on local regions of mismatches hypothesized by the ASR system inside utterances, computed between the reference (truth) and the N-best lists. Although from the formulation it is clear that this needs to be done in order to minimize the misclassification function, the global effect of the updated N-grams on the language model also needs to be considered. There can be some updates for which the overall performance (WER and PPL) is worse i.e., N-grams are updated individually to reflect/correct local erroneous regions and the interaction of these updates with other N-grams in the LM is essentially ignored. This issue has been addressed in discriminative training of acoustic models to ensure that the discriminatively trained models do not deviate too much from maximum likelihood (ML) trained models [7].

One method that imposes such a constraint calculates the distance (divergence) of the initial LM from the final updated LM. A standard measure of divergence between distributions is *relative entropy* or *Kullback-Leibler distance* which has been successfully used in the literature for pruning language models [8], and in text selection methods for LM adaptation [9]. The same technique is used here for bounding/controlling the updates. In every iteration of discriminative training, the divergence is calculated as follows

$$D(q_j||q_{j+1}) = \sum_{w_i, h_l} q_j(w_i, h_l) \log \left(\frac{q_j(w_i|h_l)}{q_{j+1}(w_i|h_l)} \right) \qquad (6)$$

where j is the iteration index, q_j is the language model obtained during the j-th step of the process and q_{j+1} denotes the models at the end of $j+1$-th iteration. The divergence is obtained by summing over all the words w_i and histories h_l.

In this work, we propose to select only those N-gram updates that minimize $D(q_j||q_{j+1})$. However, it would not be computationally feasible to minimize over all possible subsets (combinations) of N-gram updates. Instead, we assume that the N-grams affect the relative entropy roughly independently, and compute $D(q_j||q_{j+1})$ for each N-gram update. A threshold is selected and N-gram updates are pruned based on that threshold. This threshold is selected using a held-out set, analogous to minimizing perplexity on a held-out set.

To obtain the closed form solution for $D(q_j||q_{j+1})$ after applying individual N-gram updates, consider the case where q_{j+1} models are obtained from q_j by updating the N-gram, (w_x, h) (As discussed earlier, since we back-off to existing N-grams it is guaranteed $w_x \in S(h)$ is a valid assumption). We now have:

$$q_{j+1}(w_i|h) = \begin{cases} \frac{q_j(w_x|h).e^{\delta(w_x|h)}}{1+\Delta(w_x|h)} & w_i = w_x \\ \frac{q_j(w_i|h)}{1+\Delta(w_x|h)} & w_i \in S(h), w_i \neq w_x \\ \frac{\beta(h)}{1+\Delta(w_x|h)} q_j(w_i|h') & w_i \notin S(h) \end{cases} \qquad (7)$$

where $q_j(w_x|h).e^{\delta}(w_x|h) = q_j(w_x|h) + \Delta(w_x|h)$. Here, $\Delta(w_x|h)$ is the update for (w_x, h).

Now plugging Equation 7 in Equation 6, it is easy to show:

$$\begin{aligned} &D(q_j||q_{j+1}) = \\ &q_j(h) \left\{ \sum_{w_i \in S(h), w_i \neq w_x} q_j(w_i|h) \log(1 + \Delta(w_x|h)) \right. \\ &+ q_j(w_x|h) \log \left(\frac{1 + \Delta(w_x|h)}{e^{\delta(w_x|h)}} \right) + \\ &\left. \sum_{w_i \notin S(h)} q_j(w_i|h) \log(1 + \Delta(w_x|h)) \right\} \\ &= q_j(h) \left[\log(1 + \Delta(w_x|h)) - q_j(w_x|h).\delta(w_x|h) \right] \end{aligned} \qquad (8)$$

Using a threshold (which can be determined on a held-out data set) on the divergence computed from the above equation, we select the final set of N-grams to be updated.

It should be also mentioned that Equation 6 is defined only if q_j and q_{j+1} are valid probability distributions. Therefore, normalization is a necessary step for calculating the relative entropy.

IV. Language Model Adaptation

Language Model Adaptation is crucial when the training data does not match the test data being decoded. This is a frequent scenario for all ASR systems. The application domain very often contains named entities and N-gram sequences that are unique to the domain of interest. For example, conversational speech has a very different structure than classroom lectures. Linear Interpolation based methods are most commonly used to adapt LMs to a new domain. As explained in [10], linear interpolation is a special case of Maximum A Posterior (MAP) estimation, where an N-gram LM is built on the adaptation data from the new domain and the two LMs are combined using:

$$p(w_i|h) = \lambda p_B(w_i|h) + (1-\lambda)p_A(w_i|h)$$

where p_B refers to background models and p_A is the adaptation models. Also, λ is calculated by optimizing PPL/WER using the held-out data from target domain.

In the paper, we compare discriminative LM adaptation to linear interpolation and also report on their additive behavior. We propose a two-step approach. In the first step, the background models (p_B) are discriminatively trained using speech from the target domain. In the second step, we interpolate the discriminatively trained LM with the target specific LM (p_A). The motivation is to first redistribute the probabilities (by discriminative training) of the initial LM to better represent the target domain by compensating for the confusions via discriminative training.

V. Experimental Setup

The LVCSR system used throughout this paper is based on the 2007 IBM Speech transcription system for GALE Distillation Go/No-go Evaluation [11]. The acoustic models used in this system are state-of-the-art discriminatively trained models and are the same ones used for all experiments presented in this paper.

As a demonstration of the framework, we first present discriminative training results on the Hub4 portion (which corresponds to the acoustic training transcripts). The Hub4 acoustic training data is split into two sets of 350 hours and 50 hours. The initial LM is built on the 350 hour set and is subsequently discriminatively trained on the remaining 50 hour set. Discriminatively trained LMs built with the proposed method that includes *normalization* and *relative entropy based constraint*, and without the constraints are analyzed. The threshold (As discussed in Section III) for determining the validity of an updated is determined on a development set which contains both *DEV04F* and *RT03* data sets are used.

For LM adaptation experiments, the background LM (p_B, Broadcast News LM) training text consists of 335M words from the following *broadcast news* (BN) data sources [11]: 1996 CSR Hub4 Language Model data, EARS BN03 closed captions, GALE Phase 2 Distillation GNG Evaluation Supplemental Multilingual data, Hub4 acoustic model training transcripts, TDT4 closed captions, TDT4 newswire, and GALE Broadcast Conversations and GALE Broadcast News. This language model is of order 4-gram with Kneser-Ney smoothing and contains $4.6M$ n-grams based on a lexicon size of $84K$.

The second source of data is the MIT lectures data set [12] (176K words, 21 hours, 20 lectures given by two speakers). This serves as the target domain set for language model adaptation experiments. This set is split into an adaptation set comprising of 16 hours for use in discriminative training and interpolation experiments, a 2.5 hour set for evaluation and 2.5 hour set for development. The N-best list for discriminative training is generated on the 16 hours of MIT lecture data. The average N over all utterances in our experiments was determined to be 130 after careful pre-filtering of the list to remove silence and sentence-boundary markers. The out-of-vocabulary (OOV) rate using the source (BN) domain lexicon is about 1.65% on the target domain discriminative training data (16 hour set). However, acoustic scores obtained from reference alignments are needed for the discriminative training. Given the many OOV terms (acoustic scores can not be calculated for OOV terms due to the lack of pronunciation for those terms), the reference (truth) is substituted with the oracle path in the lattice and serves as a sloppy reference during discriminative training. The oracle WER of the initial lattices (using BN LM) is 14.1%.

The results are discussed in the next section.

VI. Results

The discriminative framework presented in this paper was first tested using the Hub4 experiments described in the previous section. The baseline performance is obtained using the LM built on 350 hour set. N-best lists were obtained by decoding the 50 hour set with this LM and subsequently discriminative training the LM.

It can be seen from Table I that while constrained discriminative training results in 0.2% absolute improvement over the baseline on the $RT04$ eval set, the performance gets worse using the unconstrained version.

TABLE I
$WER(\%)$ OF DISCRIMINATIVELY TRAINED LMS ON HUB4 EXPERIMENTS USING BOTH CONSTRAINED AND UNCONSTRAINED METHODS

LM	$RT04$	DEV
350 hour-LM	19.3	15.3
Disc. Train w/o Norm. and R.E	20.1	16.0
Disc. Train w/ Norm and R.E	19.1	15.0

Table II presents the results of discriminatively training the BN LM on a new domain, namely the MIT lectures. The performance of the baseline LM (BN-LM) on the evaluation and development test set is 24.7%. The BN LM is interpolated with a domain-specific LM, i.e., an LM built on the data from the MIT lecture training set (MIT-LM) and the

weights are optimized on the development test set described in Section V. This results in a WER if 17.9% and 18.5% on the development and evaluation test sets respectively (Interp.-LM line on the same Table). One-pass discriminative training comprising of 3 iterations on the initial N-best list yields a 2% absolute reduction in WER, bringing down the WER on the development and evaluation test sets to 22.4% and 22.7% respectively. A second-pass of discriminative training after regenerating lattices using the discriminatively trained LM from the first-pass lowers the WER on the development set by another 0.3%, but provides lesser reduction on the evaluation test set (0.1% reduction in WER). It can be seen from the table that the training WER, i.e. on the data used for discriminatively updating the N-grams, decreases steadily with each pass. When the discriminatively-trained LM is interpolated with the MIT-LM, the best performance is achieved with WERs of 17.5% and 18.2% respectively. This amounts to a reduction in WER of approximately 2-3% relative. It is encouraging to note that the gains from linear interpolation and discriminative training are indeed additive, albeit not by much. As described in the previous section, the oracle path from the lattices on the training data (16 hour set) is used as a reference. Table III illustrates that the oracle WER on the training data is also reduced with the discriminative training framework introduced in this paper.

TABLE II
$WER(\%)$ FOR DISCRIMINATIVELY TRAINED AND INTERPOLATED LMS ON THE TRAINING, DEVELOPMENT AND EVALUATION TEST SETS

Language Model	Training	Dev	Eval
BN-LM (%)	24.3	24.7	24.7
Interp.-LM(%)	-	17.9	18.5
Disc. Train-First Pass (%)	18.6	22.4	22.7
Disc. Train-Second Pass (%)	16.7	22.1	22.6
Disc.Train-First Pass+16 hour LM(%)	-	**17.5**	**18.2**
Disc.Train-Second Pass+16 hour LM(%)	-	17.5	18.2

TABLE III
ORACLE WER(%) OF THE BASELINE AND DISCRIMINATIVELY TRAINED LMS ON THE TRAINING DATA

LM	$WER(\%)$
BN-LM	14.1
Disc. Train-First Pass	12.5
Disc. Train-Second Pass	12.4

To analyze the effect of the three constraints introduced in this paper, namely, back-off n-gram updates, normalization and relative-entropy based global constraints, we present the decrease in WER for each training iteration in Figure 2. It can be seen from Figure 2, that the training WER decreases smoothly with each iteration when using global constraints. Figure 2 (a) and (b) indicate that although the objective function, i.e. the misclassification function (d in Equation 2), decreases in both cases, for the original framework with no constraints the training WER increases with each iteration. This can be attributed to the fact that too many simultaneous N-gram updates (of length 1 to N) with no overall global constraint on the movement of these updates, causes random fluctuations and an inconsistent set of updates.

The thresholds when applying the global constraint was chosen to be 10^{-7} for the first pass and 10^{-8} for the second pass (the thresholds are selected using the Dev set). Using these thresholds, 13% and 24% of the updates removed/pruned during the first and the second pass of discriminative training, respectively.

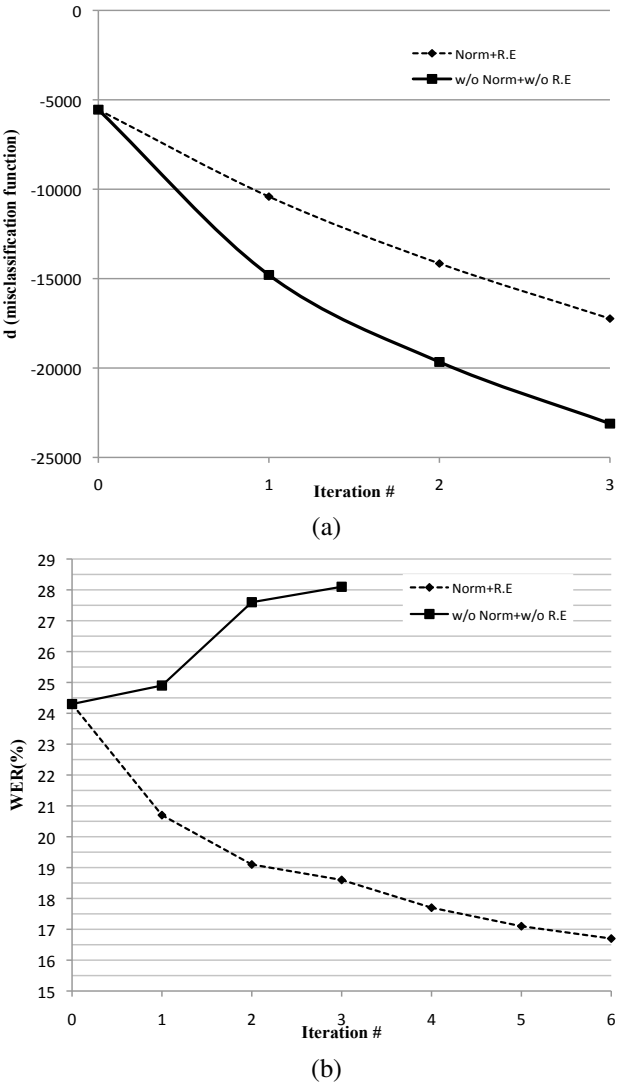

Fig. 2. Result on the training :(a) sum of d (misclassification function) over all utterances (b) $WER(\%)$ for different iterations using both methods

Table IV shows the overlap between N-grams of lattices produced on the MIT Evaluation data set with the unadapted LM (Broadcast news LM) and those N-grams which are updated during discriminative adaptation. The increased number of updates that stems from the updates to the back-off N-grams, reenforces the need for a global constraint on the movement of these updates.

TABLE IV
OVERLAP BETWEEN EVAL LATTICE N-GRAMS AND DISCRIMINATIVE UPDATES

N-gram Updates	Overlap w/ Eval N-grams(%)
Regular Updates	6.23
Back-off Updates	30.45

VII. CONCLUSION

We have introduced a framework for discriminative training of language models with constraints on the updates to the back-off n-grams and an overall global constraint on the updates. The following key points summarize this work:

- Global constraints and normalization allow for a smooth set of N-gram updates
- Discriminative training on out-of-domain data serves as an adaptation method and the gains can be further improved when such an LM is interpolated with an LM built on the out-of-domain data.

The overall performance improvements obtained are modest and additive to standard linear-interpolation methods.

VIII. ACKNOWLEDGMENT

The first author would like to thank the IBM T.J. Watson Research Lab for supporting this work as part of his internship. The Authors also want to thank Hong-Kwang Jeff Kuo for his insightful discussions and suggestions.

REFERENCES

[1] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proc. of ACL*, 1996, pp. 310–318.
[2] B. Roark, M. Saraclar, and M. Collins, "Discriminative n-gram language modeling," *Comput. Speech Lang.*, vol. 21, no. 2, pp. 373–392, 2007.
[3] H.-K. J. Kuo, E. Fosler-Lussier, H. Jiang, and C.-H. Lee, "Discriminative training of language models for speech recognition," in *Proc. ICASSP*, vol. 1, 2002, pp. 325–328.
[4] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *ACL*, 2004, pp. 47–54.
[5] A. Rastrow, A. Sethy, and B. Ramabhadran, "A new method for OOV detection using hybrid word/fragment system," in *Proc. ICASSP*, 2009.
[6] S. Katagiri, B. Juang, and C. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proceedings of the IEEE*, vol. 86, pp. 2345–2372, 1998.
[7] P. C. Woodland and D. Povey, "Large scale discriminative training for speech recognition," in *Proc.s of International Workshop on Automatic Speech Recognition*, 2000.
[8] A. Stolcke, "Entropy-based pruning of backoff language models," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.
[9] A. Sethy, S. Narayanan, and B. Ramabhadran, "Data driven approach for language model adaptation using stepwise relative entropy minimization," in *Proc. ICASSP*, 2007, pp. 177–180.
[10] M. Bacchiani, B. Roark, and M. Saraclar, "Unsupervised language model adaptation," in *Proc. ICASSP*, 2003, pp. 224–227.
[11] S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 1596–1608, 2006.
[12] J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in MIT spoken lecture processing project," in *Proc. Interspeech*, 2007.