DISCRIMINATIVE TRAINING OF CONTEXT-DEPENDENT LANGUAGE MODEL SCALING FACTORS AND INTERPOLATION WEIGHTS

S. Chang, A. Lahiri, I. Alphonso, B. Oğuz, M. Levit — *B. Dumoulin*[*]

Microsoft Corporation — Facebook Inc.

ABSTRACT

We demonstrate how context-dependent language model scaling factors and interpolation weights can be unified in a single formulation where free parameters are discriminatively trained using linear and non-linear optimization. Objective functions of the optimization are defined based on pairs of superior and inferior recognition hypotheses and correlate well with recognition error metrics. Experiments on a large, real world application demonstrated the effectiveness of the solution in significantly reducing recognition errors, by leveraging the benefits of both context-dependent weighting and discriminative training.

Index Terms— discriminative training, language model factor, interpolation, context dependent

1. INTRODUCTION

Discriminative training has long been a critical part of modern ASR, particularly for training acoustic models (AMs) (e.g. [1, 2, 3]). Even though the dominant language model (LM) training strategy remains maximum-likelihood (ML) based, there have also been many studies on discriminative LM training that aimed at reducing recognition error metrics rather than just minimizing perplexities of textual training data. Some of the studies proposed direct modification of N-gram models used in first-pass recognition (e.g. [4, 5, 6, 7]) while others focused on discriminatively training second-pass rescoring LMs that can take on more flexible structures and incorporate features beyond those generated from N-grams (e.g. [8, 9, 10]). In this work we attempt to bring the benefit of discriminative training to two additional areas in language modeling for ASR: the optimization of context-dependent LM scaling factors and context-dependent LM interpolation weights, in a unified framework.

In a typical ASR system, the total score used to rank among recognition hypotheses is a combination of scores from AM and LM. When AM and LM scores are represented as probabilities in the log domain, the combination is simply a sum. Since the estimated AM and LM scores often deviate from their ”true” values and a significant mismatch exists between the dynamic ranges of AM and LM scores, an LM scaling factor (LMF) is multiplied with LM score to improve the overall accuracy of decoding [11]. Most ASRs rely on a single global setting of LMF and its value can be set by sweeping on tuning data. However, studies have found that the optimal value of LMF may be dependent on the context and allowing LMF to be context-specific can potentially improve performance [12, 13].

In many practical scenarios the LM is actually an interpolation of multiple individual models. Some may have been built to cover different domains and styles; some may have entirely different structures, such as N-gram models and continuous-space models like neural network based LMs [14, 15]. Linear interpolation has been the most common technique due to its simplicity, and interpolation weights have traditionally been estimated using ML-based techniques. There has also been some success in optimizing interpolation weights discriminatively such as in [16]. Several studies have found that model interpolation can also be improved if interpolation weights are made context-dependent. This has been demonstrated on both word-based models [17, 18, 19] and class-based models [20].

As explained in the following sections, it is possible to design a single set of context-dependent LM weight parameters that capture the utilities of both LMF and interpolation weights. With appropriate error metrics we can discriminatively optimize the parameters on transcribed training data. The unified formulation can potentially be extended to include optimization of N-gram probabilities from the LMs as well.

Our focus on discriminative training and context-dependent parameters was motivated by the need to improve recognition accuracy of large, practical ASR applications such as personal assistant and dictation on mobile devices. The resource and real-time constraints make it difficult to incorporate into first-pass decoding certain LMs and their combinations, such as models of very large size, non-N-gram models, or those targeting diverse scenarios. To stay close to a realistic production setup, this work describes the algorithms and experimental results in a second-pass rescoring setting where we assume some first-pass recognition output is available for training and testing. It should be understood that the proposed methods can potentially be employed in first-pass recognition

[*] The author performed the work while at Microsoft Corporation

with appropriate modifications and additional engineering work to the decoder.

In the following sections, we first outline the unified formulation of context-dependent LMF and interpolation weights, followed by descriptions of discriminative training using linear and non-linear solvers. We then present experimental results and conclude with potential future work.

2. A UNIFIED FORMULATION OF CONTEXT-DEPENDENT WEIGHTS

In ASR the decoder attempts to find the optimal word sequence $\tilde{W}$ given a speech utterance X based on the maximum *a posteriori* decision rule:

$$\tilde{W} = \arg\max_{W} P(X|W)P(W) \tag{1}$$

where $P(X|W)$ is the AM likelihood function and $P(W)$ is the LM probability. Introducing the LMF λ, in log domain, the total score S for ranking alternate recognition hypotheses becomes:

$$S = \log P(X|W) + \lambda \sum_{i=1}^{|W|} \log P(w_i|h_i) \tag{2}$$

where $P(w_i|h_i)$ is the probability of word w_i given its history h_i in $W = (w_1, w_2, ...)$.

Consider a situation where the desired LM is a log-linear interpolation of a set of M individual LMs. By allowing λ to vary depending on the history h_i, we arrive at a unified formulation of context-dependent LMF and model interpolation weights:

$$S = \log P(X|W) + \sum_{i=1}^{|W|} \sum_{m=1}^{M} \lambda_m(h_i) \log P_m(w_i|h_i) \tag{3}$$

If the m^{th} LM is an N-gram model of order N, the history in $P_m(w_i|h_i)$ is limited to the preceding words $h_i = (w_{i-N+1}, w_{i-N+2}, ..., w_{i-1})$. However, we are free to augment the parameter space by having λ_m depend on more than just the history h_i. λ_m may depend on the current word w_i, or other features coming from e.g. the acoustic model, or the decoder. When λ_m is dependent on the current word w_i, model training effectively modifies the N-gram log probabilities provided by each of the LMs. Normalization is not necessary when these modified N-gram log probabilities are interpolated in our framework. In Section 4 we will compare the performance between having and not having λ_m dependent on w_i.

3. DISCRIMINATIVE TRAINING

In this section we describe how to discriminatively train the set of context-dependent LM weights $\Lambda = \{\lambda_m(h_i)\}, \forall m, i$ in (3) on a set of transcribed speech utterances $T = (t_1, t_2, ...)$. Assume we have available the recognition N-best list of each utterance in T, using a baseline ASR system with some first-pass AM and LM. The j^{th} hypothesis on the l^{th} utterance's N-best list contains the recognized word string W_l^j, the AM score $\log P(X_l|W_l^j)$, and the LM score $\log P_{orig}(W_l^j)$ of the hypothesis with respect to the original LM from the baseline decoding.

To perform discriminative training, we need to define an objective function that is correlated with recognition error metrics in some fashion. In the following sections, we introduce two different objective function formulations, aligned to two separate metrics, sentence error rate (SER) and word error rate (WER). These formulations lead to two different optimization schemes, one using linear programming and the other non-linear optimization.

3.1. Linear Optimization

In the first formulation we attempt to align the objective function to SER where a hypothesis string of an utterance is considered correct only if it matches the reference transcription entirely. Consider a subset T_c of the training data T where at least one hypothesis on the N-best list matches the reference; the remaining utterances $\bar{T_c}$ do not need to be considered since no parameter optimization can improve their SER. For the l^{th} utterance in T_c, let j^* be the index of the hypothesis on its N-best list that matches the reference. To obtain a zero SER on this utterance, we would need to ensure $S_l^{j^*} > S_l^j, \forall j \neq j^*$ with S defined as in (3). Clearly, this cannot be achieved for all utterances in T_c simultaneously. Instead, we define a hinge loss function for an utterance as following:

$$L_l = [\beta - \min_{j \neq j^*}(S_l^{j^*} - S_l^j)]^+ \tag{4}$$

where $[x]^+ = max(x, 0)$ denotes the positive part of x. This loss is essentially the margin between the scores of the correct utterance $S_l^{j^*}$, and the best competing hypothesis, capped at β and shifted so that it is non-negative. The total loss is obtained by summing the losses over all utterances in T_c:

$$L = \sum_{l=1}^{|T_c|} L_l \tag{5}$$

To minimize L with respect to Λ, we can cast it into a linear program where the objective function as well as all constraints remain linear in the free parameters Λ.

Let $\{\mu_l\}, \forall l$ be a set of slack variables, one for each utterance in T_c, to represent the loss that would incur on the utterance. The optimization task is then to minimize

$$\Omega = \sum_{l=1}^{|T_c|} \mu_l \tag{6}$$

such that,

$$S_l^{j^*} + \mu_l > S_l^j, \forall l, j \neq j^* \tag{7}$$

and

$$\mu_l > -\beta, \forall l \tag{8}$$

The linear programming problem can be solved efficiently with standard constrained optimization methods. The optimal value of β can be determined on a tuning data set.

3.2. Non-linear Optimization

In the second formulation we consider pairs of hypotheses from each utterance's N-best list. Instead of just looking at correctness at the utterance level, we compute the word-level edit distance between the reference transcription and each hypothesis of an utterance and construct pairs containing a hypothesis with the lowest edit distance and each of the remaining hypotheses with higher edit distances. This selection of hypothesis pairs is designed to discriminate between the best hypothesis and each of the worse hypotheses for an utterance in terms of word edit distance to reference, and thus aims at improving the 1-best hypothesis WER.

We would like to reward pairs where the hypothesis with lower edit distance has a higher total score S, as defined in (3), than the hypothesis with higher edit distance, and to penalize those of the opposite ordering. Let $\Psi_l = \{(j^*, j)\}$ denote a set of pairs of hypotheses for the l^{th} utterance where $W_l^{j^*}$ has the lowest edit distance among all hypotheses of this utterance and W_l^j has a higher edit distance than W_l^{j*}. For a pair (j^*, j), define a score $Q_l(j^*, j)$ as:

$$Q_l(j^*, j) = \frac{1}{1 + e^{-\alpha(S_l^{j^*} - S_l^j)}} \tag{9}$$

where α modulates the steepness of the sigmoid function and can be adjusted on a tuning data set. Aggregating over all selected pairs of hypotheses in the training data, the overall objective function that we would like to maximize is defined as:

$$\Omega = \sum_{l=1}^{|T|} \sum_{\Psi_l} \frac{1}{1 + e^{-\alpha(S_l^{j^*} - S_l^j)}} \tag{10}$$

where $(j^*, j), \forall j \neq j^*$ ranges over all selected hypothesis pairs for the l^{th} utterance as described above.

The gradient of Ω with respect to each context-dependent $\lambda_m(h_i)$ can be computed as:

$$\frac{\partial \Omega}{\partial \lambda_m(h)} = \sum_{l=1}^{|T|} \sum_{\Psi_l} \frac{\alpha e^{-\alpha(S_l^{j^*} - S_l^j)}}{(1 + e^{-\alpha(S_l^{j^*} - S_l^j)})^2} \frac{\partial (S_l^{j^*} - S_l^j)}{\partial \lambda_m(h)} \tag{11}$$

and

$$\frac{\partial (S_l^{j^*} - S_l^j)}{\partial \lambda_m(h)} = \sum_{i, \forall (h_i, w_i) \in W_l^{j^*}, h_i = h} \log P_m(w_i^{j^*} | h_i) - \sum_{i, \forall (h_i, w_i) \in W_l^{j}, h_i = h} \log P_m(w_i^{j} | h_i) \tag{12}$$

where $(h_i, w_i) \in W_l^j$ means $h_i w_i$ is a substring of W_l^j. The optimization can be performed with standard quasi-newton methods [21]. To improve generalization we also included the $L2$-norm of Λ for regularization during training.

3.3. N-gram Context Cutoff and Back-off

Defining a separate set of λs for each N-gram context generates a very large number of parameters, which can be prohibitively expensive for training, and more critically, lead to severe overtraining. Various techniques were proposed to improve the robustness of maximum likelihood context-dependent interpolation weight training [18, 20]. One technique is to apply a cutoff threshold on the minimum number of times an N-gram context must be observed in the training data to warrant a separate parameter. If an N-gram context does not make the cut, it is backed off to a lower-order context, repeatedly if necessary, until a null context is reached, in which case the context-independent λs are employed. At testing time, the same back-off scheme is applied if λs for an N-gram context are not found in the trained parameters.

However, this simple back-off scheme leads to the undertraining of weights associated with some shorter contexts if they occur mostly as a substring of a popular longer context in the training data. For example, assume in a particular training data set, most of the occurrences of the word *Obama* occurs in bigram contexts *Barack Obama*, *President Obama* and *Michelle Obama*. The weights associated with these three bigram contexts can be well trained but the weight associated with the unigram context *Obama* will not since very few training samples will back off to this unigram context. If on a test set there is an unseen bigram context *Natasha Obama*, it has to use an undertrained weight associated with the unigram context *Obama*. To remedy this situation, instead of using a strict back-off scheme, we apply an interpolated weighting:

$$\lambda(w_{i-N+1}, ..., w_i) = \sum_{n=1}^{N} \lambda'_n(w_{i-n+1}, ..., w_i) + \lambda'_0 \tag{13}$$

where λ'_0 is a context-independent weight. This scheme ensures that weights associated with all contexts surviving the frequency cutoff will be trained on sufficiently large number of samples. All our experiments with context-dependent weights in Section 4 adopt this strategy.

4. EXPERIMENTS

We tested the proposed methods on real user data collected from Cortana, Microsoft's automated personal assistant application. The utterances cover a wide range of domains such as communication, reminder, on-device search, and general web search, with a median of four words per utterance. A total of 361K utterances (TRAIN) were used for training and 44K (VALID) for tuning and validation. Another 44K (TEST) utterances containing 208K words were used for testing. All utterances were professionally transcribed. As described previously our experiments were conducted as N-best rescoring on first-pass recognition outputs, which were taken from production results where each user utterance was recognized with a conjunction of a large generic LM and smaller but personalized and contextualized LMs. The generic LM was a 5-gram LM trained from a large body of text covering all domains relevant to the application. The AM was a sequence-trained context-dependent DNN model with a front-end of 29 log-filter bank features and their first and second derivatives.

Each utterance's first-pass result includes an N-best list of up to ten hypotheses, containing recognized string, and AM and LM scores for the entire utterance. It could potentially be beneficial to include a larger N-best list for each utterance, especially for recognition scenarios with relatively long utterances. However, for our task of mostly relatively short utterances, 10-best hypotheses offer a reasonably large room for improvement through rescoring - the oracle 10-best WER is about 40% lower than the 1-best WER. The linear solver trains on a subset of the training utterances where, as described in Section 3, at least one hypothesis on the N-best list matches the reference. In our TRAIN set, this amounts to about 83% of the utterances. For non-linear solver, all utterances in TRAIN were used for training.

To test LM interpolation we collected a set of 6 models, $LM_1, LM_2, ..., LM_6$, that were not used for generating the production recognition results and were not trained on TRAIN, VALID or TEST sets. These LMs included N-gram models as well as RNN-LMs [15]; some models cover multiple domains while others focus on a narrower set of scenarios. We pre-sorted the LMs by their potential benefit for rescoring, with LM_1 being the most useful. The ordering was obtained iteratively by, starting from an empty set, greedily adding the LM that brings the most WER reduction on VALID data. Since these LMs did not necessarily cover the entire spectrum of information that the LMs in the first-pass recognition did, in all experiments, we included the first-pass LM score as an interpolation component, with a context-independent weight parameter. For non-linear optimization, we used an L-BFGS implementation from Microsoft Solver Foundation [22]. For linear optimization Gurobi's implementation of the Simplex method [23] was used. An early stopping criterion was applied to prevent overtraining when WER no longer decreased on the VALID set over a number of iterations.

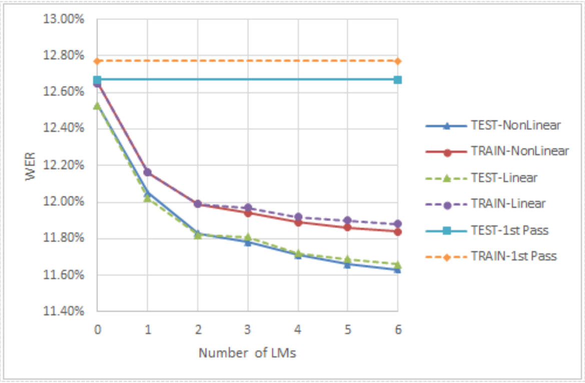

Fig. 1. *WER% comparison between linear and non-linear solvers on TEST and TRAIN, for context-independent weights.*

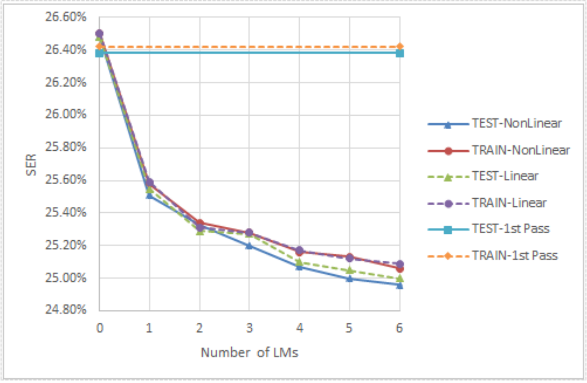

Fig. 2. *SER% comparison between linear and non-linear solvers on TEST and TRAIN, for context-independent weights.*

4.1. Discriminative Training Results

We first trained and tested context-independent interpolation weights for different number of LMs. For each of $M = 1, 2, .., 6$ the first M models $LM_1, .., LM_M$, according to the ordering as described above, were included in the interpolation set. Figure 1 compares WERs of rescoring using weights trained by linear and non-linear solvers, on TEST as well as TRAIN set. With both solvers, the TEST WER decreases as more LMs are added to the interpolation, and closely tracks the decrease in TRAIN WER. Figure 2 makes a similar comparison on SERs of rescoring using weights trained with linear and non-linear solvers. Even though the linear solver was designed to optimize a metric that correlates with SER while the non-linear solver was not, we did not observe an advantage from the linear solver over non-linear solver with respect to the TEST SERs. For brevity of explanation, we will focus on only the non-linear solver results for the remainder of the experiments.

In the next set of experiments we trained context-dependent weights for each number of LMs. The context lengths for N-gram history and cutoff threshold for context observations were tuned on VALID set. The context length was set to 2, which could include up to two preceding words, in addition to the current word, for each λ. The cutoff threshold for context observation was set to 25.

Table 1 compares TEST set WERs between context-independent and context-dependent weights, using non-linear solver for each size of the interpolation set. Note that the first row with zero LM is when only a single LMF was trained based on the first-pass LM score, without any new LMs being interpolated. For this special case we also tried sweeping the value of the single LMF to find the minimum WER on the TRAIN set, and verified that the non-linear solver solution matched the result of the parameter sweeping. The first-pass baseline WER was 12.67% and the six-model context-dependent interpolation provided 12.2% relative WER reduction over the baseline. The last column shows that discriminative training with context-dependent weights provided between 4.03% and 4.56% relative WER reduction over the context-independent weights for various number of LMs, all of which are statistically significant at the 0.01 level using a difference of proportions significance test.

Number	Non-linear Solver		WERR
of LMs	CI WER%	CD WER%	CI to CD%
0	12.53	-	-
1	12.05	11.56	4.07
2	11.83	11.29	4.56
3	11.78	11.27	4.33
4	11.71	11.19	4.44
5	11.66	11.19	4.03
6	11.63	11.13	4.30

Table 1. *Non-linear solver TEST WER% for context-independent (CI) and dependent (CD) weights, and relative WER reduction from CI to CD, for different number of LMs. The 1st-pass baseline WER was 12.67%.*

We also compared the performance of the context-dependent weights with and without including the current word in the history context for each λ. Table 2 shows that for each number of LMs in interpolation, including the current word in the history context provides a small but consistent WER reduction. The inclusion of the current word in the context history also increases the total number of the free parameters by about 30%. However, even with an average of 72K parameters per LM being interpolated, the total number of free parameters in our proposed solution is still at least two orders of magnitude smaller than the number of free parameters in the LMs themselves, which ensures the discriminative training can be carried out robustly on a modestly sized training data.

Number	No Current Word		With Current Word	
of LMs	WER%	# Parameters	WER%	# Parameters
1	11.63	53K	11.56	72K
2	11.40	106K	11.29	144K
3	11.29	156K	11.27	217K
4	11.29	213K	11.19	289K
5	11.26	266K	11.19	361K
6	11.22	319K	11.13	433K

Table 2. *Non-linear solver TEST WER% and number of parameters for context dependent (CD) weights, without and with current word included in history context for λ, for different number of LMs.*

4.2. Comparison to Alternative Uses of Training Data

Discriminative training of context-dependent weights requires a significant amount of transcribed training data. It is natural to ask how the proposed solution compares to alternatives that use the same amount of training data. Table 3 compares the performance of several alternative model interpolation methods for N-best rescoring, using the first two models in our collection, LM_1 and LM_2, which are both 5-gram ARPA LMs. For each method, we compare WER, SER and relative reductions (WERR and SERR) with respect to *1st Pass Baseline*, on TEST set.

1. *2Model ML-CI*: Instead of using the two LMs directly in rescoring, we first merged them offline using context-independent linear interpolation weights estimated in an ML fashion by running EM iterations to minimize perplexities of TRAIN data;

2. *2Model ML-CD*: Same as 1. except using context-dependent linear interpolation, similar to methods in [18];

3. *2Model Disc CI*: Discriminatively trained context-independent interpolation, same as the 2-model performance of TEST-NonLinear in Figure 1;

4. *2Model Disc CD*: Our proposed method of discriminative training of context-dependent log-linear interpolation, optimized with non-linear solver.

5. *2Model+TR-LM*: Instead of using TRAIN set for interpolation, we trained a 5-gram LM (TR-LM) from the TRAIN set, and included it in rescoring, along with LM_1 and LM_2;

For 1. and 2., since TRAIN set had already been used to merge LMs offline, we used VALID set utterances to estimate the context-independent interpolation weights between the merged LM and the 1st-pass LM score for N-best rescoring, using the same discriminative training method in Section 3.2. Similarly, for 5., since TRAIN set had been used for training TR-LM, we again used VALID set utterances for estimating context-independent weights for rescoring. The use of VALID set for training in those three alternatives actually gives them a slight advantage over *2Model Disc CD*. Nevertheless, our proposed method still outperforms the alternatives. In particular, discriminative training provides significant gain over ML training when both were using context-dependent weights; and, context-dependent interpolation significantly improves over context-independent interpolation when both were trained discriminatively. The proposed solution also provides a more efficient use of TRAIN data compared to training a separate TR-LM as in 5. The WER reduction between our proposed solution and each of the alternatives was statistically significant at the 0.01 level using a difference of proportions significance test.

	WER%	SER%	WERR%	SERR %
1st Pass Baseline	*12.67*	*26.38*	-	-
2Model ML-CI	11.86	25.29	6.39	4.13
2Model ML-CD	11.83	25.17	6.63	4.59
2Model Disc CI	11.83	25.33	6.63	3.98
2Model Disc CD	**11.29**	**24.51**	10.89	7.09
2Model+TR-LM	11.64	25.07	8.13	4.97

Table 3. *Comparing TEST set WER, SER and relative changes with respect to 1st Pass Baseline, for alternative strategies of interpolating two models.*

5. CONCLUSION AND FUTURE WORK

We presented a unified formulation for context-dependent language model scale factors and interpolation weights, and developed a discriminative training technique to optimize the parameters, using linear or non-linear optimization. The experimental results of N-best rescoring on a large, real world application demonstrated the effectiveness of the proposed solution.When up to six language models were interpolated with discriminatively trained context-dependent weights, we obtained over 12% relative reduction in WER over the first-pass baseline. In particular, context-dependent weights provided up to 4.6% additional WER reduction over their context-independent counterparts, which is comparable to some of the best improvements that discriminative LM training was reported to have achieved in previous works. We also demonstrated that the proposed solution outperformed several alternative methods for interpolating models using same amount of training data.

There are multiple directions to further improve the solution. First, scalability of training requires further improvement to support large parameter sets and training data. Techniques such as parameter tying and training data selection may potentially improve the efficiency of the parameters and reduce overtraining. Second, the unified formulation in (3) can be extended to include additional contextual information beyond the N-gram context, for example, information about acoustic conditions. Finally, the requirement of having a large set of transcribed training data increases the cost of the proposed solution. Some previous studies have successfully applied unsupervised or semi-supervised strategies to increase the amount of training data for discriminative LM training at relatively low cost (e.g. [24, 25, 26]). Our proposed solution can be easily extended to train on unsupervised or semi-supervised materials that are far more abundant than manual transcriptions, as long as some notion of superiority between pairs of hypotheses can be defined.

6. REFERENCES

[1] L. Bahl, P. Brown, P. De Souza, and R. Mercer, “Maximum mutual information estimation of hidden Markov model parameters for speech recognition,” in *Proc. of ICASSP*. IEEE, 1986, vol. 11, pp. 49–52.

[2] P.C. Woodland and D. Povey, “Large scale discriminative training of hidden Markov models for speech recognition,” in *Computer Speech and Language*, 2002, vol. 16, pp. 25–47.

[3] D. Povey and P.C. Woodland, “Minimum phone error and I-smoothing for improved discriminative training,” in *Proc. of ICASSP*. IEEE, 2004.

[4] J. Huang, X. Li, and A. Acero, “Discriminative training methods for language models using conditional entropy criteria,” in *Proc. of ICASSP*. IEEE, 2010.

[5] J. Kuo, E. Fosler-Lussier, H. Jiang, and C. Lee, “Discriminative training of language models for speech recognition,” in *Proc. of ICASSP*. IEEE, 2002.

[6] V. Magdin and H. Jiang, “Discriminative training of N-gram language models for speech recognition via linear programming,” in *Proc. of ASRU*. IEEE, 2009.

[7] A. Rastrow, A. Sethy, and B. Ramabhadran, “Constrained discriminative training of N-gram language models,” in *Proc. of ASRU*. IEEE, 2009.

[8] P. Jyothi, L. Johnson, C. Chelba, and B. Strope, “Distributed discriminative language models for Google Voice Search,” in *Proc. of ICASSP*. IEEE, 2012.

[9] B. Roark, M. Saraclar, M. Collins, and M. Johnson, “Discriminative language modeling with conditional random fields and the perceptron algorithm,” in *Proc. of ACL*. ACL, 2004.

[10] 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. of ICASSP*. IEEE, 2006.

[11] K.-F. Lee, *The Development of the SPHINX System*, Kluwer Academic Publishers, Boston, 1989.

[12] X. Huang, M. Belin, F. Alleva, and M. Hwang, "Unified stochastic engine (USE) for speech recognition," in *Proc. of ICASSP*. IEEE, 1993.

[13] B. Hoffmeister, R. Liang, R. Schluter, and H. Ney, "Log-linear model combination with word-dependent scaling factors," in *Proc. of Interspeech*. ISCA, 2009.

[14] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Journal of Machine Learning Research*, 2003, pp. 3:1137–1155.

[15] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. of Interspeech*. ISCA, 2010.

[16] P. Beyerlein, "Discriminative model combination," in *Proc. of ICASSP*. IEEE, May 1998.

[17] B. Hsu, "Generalized linear interpolation of language models," in *Proc. of ASRU*. IEEE, 2007.

[18] X. Liu, M.J.F. Gales, and P.C. Woodland, "Context dependent language model adaptation," in *Proc. of Interspeech*. ISCA, 2008.

[19] I. Bulyko, M. Ostendorf, and A. Stolcke, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures," in *Proc. of HLT-NAACL*. ACL, 2003.

[20] M. Levit, A. Stolcke, S. Chang, and S. Parthasarathy, "Token-level interpolation for class-based language models," in *Proc. of ICASSP*. IEEE, 2015.

[21] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[22] Microsoft Solver Foundation, *https://msdn.microsoft.com/en-us/library/ff524509(v=vs.93).aspx*.

[23] Gurobi Optimization, *http://www.gurobi.com/*.

[24] A. Celebi et al., "Semi-supervised discriminative language modeling for Turkish ASR," in *Proc. of ICASSP*. IEEE, 2012.

[25] P. Xu, D. Karakos, and S. Khudanpur, "Self-supervised discriminative training of statistical language models," in *Proc. of ASRU*. IEEE, 2009.

[26] E. Dikici and M. Saraclar, "Unsupervised training methods for discriminative language modeling," in *Proc. of Interspeech*. ISCA, 2014.