Minimum Bayes Risk Discriminative Language Models for Arabic Speech Recognition

Hong-Kwang Jeff Kuo, Ebru Arısoy, Lidia Mangu, George Saon

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, U. S. A.
{hkuo, earisoy, mangu, gsaon}@us.ibm.com

***Abstract*—In this paper we explore discriminative language modeling (DLM) on highly optimized state-of-the-art large vocabulary Arabic broadcast speech recognition systems used for the Phase 5 DARPA GALE Evaluation. In particular, we study in detail a minimum Bayes risk (MBR) criterion for DLM. MBR training outperforms perceptron training. Interestingly, we found that our DLMs generalized to mismatched conditions, such as using a different acoustic model during testing. We also examine the interesting problem of unsupervised DLM training using a Bayes risk metric as a surrogate for word error rate (WER). In some experiments, we were able to obtain about half of the gain of the supervised DLM.**

I. Introduction

DLM (discriminative language model) is a log-linear model complementary to the existing generative n-gram language model (LM). In contrast to the generative model, it is trained on patterns of confusions or errors made by the speech recognizer to optimize an objective function that is directly related to the word error rate. Thus the model learns from both positive examples (correct transcriptions) and negative examples (recognition errors).

Discriminatively trained LMs have been demonstrated to consistently outperform generative LMs, partly due to improved parameter estimation and partly due to the ease with which many arbitrary features can be included into the same model. Linear models have been successfully applied to DLMs for speech recognition. The perceptron algorithm [1], [2] or methods based on maximizing the conditional log-likelihood [1] or minimizing a discriminative loss function [3] have been used for DLM parameter estimation. A comparison of different loss functions for DLMs is given in [4]. In addition to discriminative linear models, a minimum classification error framework was used for adjusting the parameters of generative n-gram LMs to achieve minimum sentence error [5], [6].

In DLM based on linear models, hypothesis sentences are represented as a bundle of arbitrary features, and feature parameters are estimated discriminatively. The traditional approach is to use word n-gram counts as DLM features. Being a feature-based language modeling approach allows the DLM to integrate relevant information sources into LMs. Therefore part-of-speech tag features [7], [8], parse tree features [7], [8], morphological features [9], [10], discourse level trigger features [11], automatically induced sub-word class features [8] and word class features [12], and state and duration features [13] extracted from the clustered allophone state sequences have been explored within the DLM paradigm, in addition to basic word n-gram features.

In this paper we explore using DLMs to improve our best Arabic speech recognition systems for the latest Phase 5 DARPA GALE Evaluation. Besides the well known parameter estimation methods, perceptron and global conditional log-linear model (GCLM) [1], we propose a minimum Bayes risk (MBR) objective for DLM. MBR has been used in ASR decoding [14], [15], [16] and model combination [17], but not extensively in discriminative language modeling. Compared to previous published results, our experiments are noteworthy for being applied to highly optimized state-of-the-art systems that include speaker adaptation, feature and model space discriminative training, and even sophisticated new models like Bayesian sensing [18], [19], [20]. The systems have been trained on large amounts of data, including 1800 h of acoustic data and 1.6 billion words, and attain average WERs of about 10%. In addition, the baseline LMs are already very strong, including neural network LM (NNLM) [21], [22] and model M [23], a class-based exponential LM.

Our goal was to apply DLMs to improve the DARPA Evaluation results. Multiple acoustic models were continually being improved until the deadline, so there was no time to use the best final acoustic models to decode the DLM training data. Therefore an important question we address is: can a DLM trained on some acoustic model be applied when decoding with a different acoustic model?

We also investigate a new interesting problem of using un-transcribed speech data for DLM training. In the real world, we are likely to have much more un-transcribed audio data and text data than transcribed audio data. To take advantage of large amounts of text data without corresponding audio, [24], [25] generate negative examples for DLM training by simulating ASR errors for the text data, after having learned phone confusion patterns from ASR outputs of transcribed data. In contrast, to take advantage of un-transcribed audio data, we explore unsupervised DLM training by using a Bayes risk metric as a surrogate for WER. After any speech application has been deployed, a large amount of un-transcribed audio data can be collected, and this data is arguably more relevant and in-domain than the available text data.

Section II reviews the formulation for DLM, and Section III describes parameter estimation methods for DLM, one of which is based on minimum Bayes risk. Section IV discusses

unsupervised DLM training. Section V describes our experimental setup. Section VI presents the experimental results, and we conclude with a summary in Section VII.

II. Discriminative LM

The main components of the DLM are the model parameter vector $\bar{\alpha}$ and feature vector $\Phi(x, y)$ which is a function of the input acoustic data, x, and word sequence hypothesis, y. Suppose there are a total of d features. Then the parameter vector is $\bar{\alpha} = [\alpha_0 \dots \alpha_d]^T$ and the feature vector is $\Phi(x, y) = [\phi_0(x, y) \dots \phi_d(x, y)]^T$. Typically the first component of the feature vector $\phi_0(x, y)$ comes from the speech recognizer, either a weighted combination of acoustic and language model scores [1], or the log posterior probability obtained by normalizing the weighted score over the lattice produced for x. The other components of the feature vector are typically n-gram counts, i.e. the number of times a particular n-gram is seen in the candidate hypothesis y. An example word bigram feature is as follows.

$\phi_i(x, y)$ = the number of times "`the paper`" is seen in the candidate hypothesis y

DLM allows for easy integration of relevant information sources, such as morphology, syntax and semantics, into the LM via features. In this paper we mainly focus on parameter estimation algorithms rather than investigating different types of features in Arabic DLM. For highly inflectional language (of which Arabic is one), sub-word n-gram features have been shown to yield higher improvements than word n-gram features [8]. Therefore we used morph n-gram features. One advantage over words is a smaller vocabulary size.

The model parameter vector $\bar{\alpha}$ is estimated during DLM training using the training examples (x_i, y_i) for $i = 1 \dots I$. Here y_i is the reference transcript corresponding to the acoustic input x_i. From the acoustic input x_i, the $\mathbf{GEN}(x_i)$ function enumerates a finite set of candidates for DLM training. The $\mathbf{GEN}(x_i)$ function can be the lattice or the N-best list output of the baseline ASR system for the utterance x_i. In this paper we used N-best lists as candidate hypotheses. After each candidate hypothesis is mapped to an d-dimensional feature vector, the feature parameters, α_i's, can be learned discriminatively [1], [2], [3].

The posterior probability of a hypothesis y for an input x_i under the DLM parameters $\bar{\alpha}$ is defined to be

$$p_{\bar{\alpha}}(y|x_i) = \frac{e^{\Phi(x_i,y)\cdot\bar{\alpha}}}{\sum_{y' \in \mathbf{GEN}(x_i)} e^{\Phi(x_i,y')\cdot\bar{\alpha}}} \quad (1)$$

During decoding of utterance i, the DLM chooses the word hypothesis in the N-best list with the highest score, or equivalently, highest posterior probability

$$\hat{y}_i = \text{argmax}_{y \in \mathbf{GEN}(x_i)} \Phi(x_i, y) \cdot \bar{\alpha} \quad (2)$$

[1] $\log P_{LM}(y) + \frac{1}{\beta} \log P_{AM}(x|y)$ where $\log P_{LM}(y)$, $\log P_{AM}(x|y)$ and β are the baseline LM score, acoustic model score and LM weight, respectively.

Inputs: Training examples (x_i, y_i) for $i = 1 \dots I$
Initialization: $\bar{\alpha}_0^I = (\alpha_0, 0, \dots, 0)$
Algorithm:
For $t = 1 \dots T$
  $\bar{\alpha}_t^0 = \bar{\alpha}_{t-1}^I$
  For $i = 1 \dots I$
    $\hat{y}_i = \text{argmax}_{y \in \mathbf{GEN}(x_i)} \Phi(x_i, y) \cdot \bar{\alpha}_t^{i-1}$
    $\bar{\alpha}_t^i = \bar{\alpha}_t^{i-1} + \Phi(x_i, y_i) - \Phi(x_i, \hat{y}_i)$
Output: Averaged parameters $\bar{\alpha} = \sum_{i,t} \bar{\alpha}_t^i / IT$

Fig. 1. Perceptron algorithm. $\bar{\alpha}_t^i$ represents the feature parameters after the t'th pass on the i'th example. y_i is the gold-standard hypothesis.

III. Parameter Estimation for DLM

In this section we revisit the perceptron algorithm [1], [2] and global conditional log-linear model (GCLM) [1]. We also explain in detail the minimum Bayes risk (MBR) objective function for DLM parameter estimation.

A. Perceptron Algorithm

A popular algorithm for DLM parameter estimation is the perceptron algorithm, shown in Figure 1. At the beginning of the algorithm, all the feature parameters except α_0 (weight associated with $\phi_0(x, y)$) are initialized with 0. For each example x_i, the best hypothesis under the current model is chosen to maximize the inner product of the feature and parameter vectors. The parameters, except α_0, are updated to penalize features associated with the current 1-best hypothesis, and to reward features associated with the gold-standard hypothesis (reference or lowest-WER hypothesis). It has been found that the perceptron model trained with the reference transcript as the gold-standard hypothesis is more sensitive to the value of the α_0 constant [1]. Instead, we use the oracle hypothesis as the gold-standard hypothesis. Averaged parameters are used in decoding held-out and test sets.

B. Global Conditional Log-Linear Model

Another popular algorithm uses the GCLM, where the objective is to maximize the conditional log-likelihood of the training data under the parameters $\bar{\alpha}$. The objective function of GCLM is as follows:

$$F = \sum_{i=1}^{I} \log p_{\bar{\alpha}}(y_i|x_i) = \sum_{i=1}^{I} \log \frac{e^{\bar{\alpha}\cdot\Phi(x_i,y_i)}}{\sum_{y \in \mathbf{GEN}(x_i)} e^{\bar{\alpha}\cdot\Phi(x_i,y)}}. \quad (3)$$

The numerator can be thought of as the score of the correct hypothesis while the denominator is a sum of the scores of all N-best hypotheses. F is a convex function so the optimal parameters can be found with simple gradient updates:

$$\bar{\alpha}' = \bar{\alpha} + \epsilon \sum_{i=1}^{I} \left(\Phi(x_i, y_i) - \sum_{y \in \mathbf{GEN}(x_i)} p_{\bar{\alpha}}(y|x_i) \Phi(x_i, y) \right) \quad (4)$$

Thus while the perceptron algorithm compares only the reference hypothesis and the 1-best hypothesis, GCLM considers all N-best hypotheses in the parameter estimation. The perceptron algorithm is typically used for feature selection

and model initialization, and from that starting point, GCLM training has been shown to further improve the model [1]. Intuitively, Equation 4 says that for a feature that does not appear in the reference, subtract the posterior probability of the feature. For a feature that does appear in the reference, the update is the difference between 1 and the posterior probability. Hence if the reference hypothesis already has a large sentence posterior probability, then the update will be small. In this paper we used a regularized objective function with zero mean Gaussian prior ($||\bar{\alpha}||^2/2\sigma^2$).

C. MBR objective function

We define the minimum Bayes risk (MBR) objective function, where we try to minimize the expected loss

$$F = \frac{1}{N_c}\sum_{i=1}^{I} E_{y|x_i}[L(y,y_i)] \quad (5)$$

$$= \frac{1}{N_c}\sum_{i=1}^{I}\sum_{y\in\mathbf{GEN}(x_i)} L(y,y_i)p_{\bar{\alpha}}(y|x_i), \quad (6)$$

where N_c is the total number of words in the corpus, and $L(y,y_i)$ is the number of word errors (substitution, insertion, deletion) of hypothesis y compared with reference y_i. Note that $L(y,y_i)$ is the number of errors, not a word error rate. This mirrors the way that WER is calculated: by summing up the errors and dividing by the total number of words in the corpus. The MBR objective function can be thought of as a smooth estimate of the actual WER. It is similar to the MERT objective function [26], [27], although the interpretation may be a bit different.

We rewrite the objective function as

$$F = \frac{1}{N_c}\sum_{i=1}^{I} F_i, \quad (7)$$

where for each utterance

$$F_i = \sum_{y\in\mathbf{GEN}(x_i)} \frac{L(y,y_i)e^{\Phi(x_i,y)\cdot\bar{\alpha}}}{\sum_{y'\in\mathbf{GEN}(x_i)} e^{\Phi(x_i,y')\cdot\bar{\alpha}}}. \quad (8)$$

Then for each feature s, ($\Phi = (\phi_0 \ldots \phi_s \ldots \phi_d)$)

$$\frac{\partial F_i}{\partial \alpha_s} = -\gamma_{i,s}(l_i^{\text{avg}} - l_{i,s}), \quad (9)$$

where

$$\gamma_{i,s} = \sum_{y\in\mathbf{GEN}(x_i)} \phi_s(x_i,y)p_{\bar{\alpha}}(y|x_i) \quad (10)$$

is the expected count of feature s in sentence i,

$$l_i^{\text{avg}} = \sum_{y\in\mathbf{GEN}(x_i)} L(y,y_i)p_{\bar{\alpha}}(y|x_i) = F_i \quad (11)$$

is the expected loss of sentence i, considering all N-best hypotheses, and

$$l_{i,s} = \frac{\sum_{y\in\mathbf{GEN}(x_i)} L(y,y_i)\phi_s(x_i,y)p_{\bar{\alpha}}(y|x_i)}{\sum_{y'\in\mathbf{GEN}(x_i)} \phi_s(x_i,y')p_{\bar{\alpha}}(y'|x_i)} \quad (12)$$

is the expected loss of the sentence, considering the subset of hypotheses containing feature s. If there is only one hypothesis y^* containing s, then $l_{i,s} = L(y^*,y_i)$.

This formulation is similar to MPE training for discriminative estimation of HMM parameters for ASR [28]. There, Equation 9 is used to compute MPE arc occupancies over a word lattice. The occupancies are used to compute statistics which are used in extended Baum-Welch updates of the Gaussian means and variances.

For the DLM, intuitively, the MBR gradient update says that, for each feature, if the expected loss of the sentence, considering all N-best, is more than the expected loss, considering only hypotheses containing that feature, then the feature is "helpful," so increase the parameter. If the loss considering all N-best is less than the loss considering hypotheses with that feature, then decrease the parameter.

Notice that the nature of the updates for GCLM and MBR are intuitively different. GCLM does not take the errors of each N-best into account and explicitly uses the reference/oracle hypothesis. MBR does not explicitly use a reference/oracle hypothesis but takes advantage of the error information for each N-best hypothesis to reduce the expected loss. Not having to specify a single oracle hypothesis is useful, since there could be multiple hypotheses with the same WER after text normalization, e.g. *it is* vs. *it's*. Text normalization such as hamza normalization is very prevalent in Arabic ASR.

IV. Unsupervised DLM

DLM normally requires acoustic data with their reference transcripts as the training data. One strength of DLM lies in learning from the recognition errors as negative examples and correcting these errors during decoding. However, the amount of acoustic data with reference transcripts is very limited compared to the available text data used in generative language model training. Moreover, it is much cheaper to collect acoustic data than to have the data also transcribed by humans. There will always be a lot of un-transcribed acoustic data, and how to take advantage of such data is an important problem. In this paper, we explore using un-transcribed speech data for DLM training.

We borrow ideas from minimum Bayes risk classifiers, which have been used in ASR [14], [15] and N-best list re-scoring [16], to tackle the problem of unsupervised DLM training. A Bayes risk metric for a hypothesis y can be calculated from the N-best hypotheses as follows:

$$l(y|x_i) = E_{y'|x_i}[L(y,y')] \quad (13)$$

$$= \sum_{y'\in\mathbf{GEN}(x_i)} L(y,y')p(y'|x_i),$$

where $L(y,y')$ is the number of errors (substitution, insertion, deletion) between the hypotheses y and y' when y' is considered as the reference, and $p(y'|x_i)$ is the posterior probability of the hypothesis y' produced by the baseline recognizer for the acoustic input x_i. Given the Bayes risk, $l(y|x_i)$, for all the hypotheses in the N-best list, $\hat{y}_i$ is the minimum Bayes risk (MBR) hypothesis for x_i:

$$\hat{y}_i = \text{argmin}_{y\in\mathbf{GEN}(x_i)} l(y|x_i). \quad (14)$$

In the supervised DLM scenario the reference transcript is used to calculate the number of errors for each hypothesis to pick the oracle or to be used in MBR DLM training. In the unsupervised approach, we assign the Bayes risk metric in lieu of the number of errors to each N-best hypothesis in the training data. For each hypothesis, this metric is calculated using Eq. 13. For perceptron training, instead of the reference/oracle hypothesis, we use the MBR hypothesis. We specify the oracle hypothesis in the same way for GCLM training. For MBR DLM training, instead of using word errors for $L(y, y_i)$, we use the risk metric, $l(y|x_i)$. The update equations and the formulations remain the same.

V. Experimental Setup

Data preparation for DLM training involves the following issues and steps.

1) **Select speech recognizer.** We chose an Arabic ASR system used in Phase 3 of the DARPA GALE Evaluation. The acoustic model was trained on 1500 h of data and included speaker adaptation, and feature and model space discriminative training. It is an un-vowelized (or grapheme) system that uses pronunciations constructed by using the letters as phones without using the short vowel information encoded in diacritics (which are typically missing in written Arabic text). We label this acoustic model **P3U** (for Phase 3 **U**nvowelized). The language model was trained on 20 corpora with about 1.6 billion words, with a vocabulary size of 795K words. The LM is an interpolated Kneser-Ney smoothed 4-gram LM, pruned to 77M n-grams.
2) **Select DLM training data.** In Phase 4 we received an additional 300 h of acoustic data (about 160K sentences and 2.5M words) with no overlap with the 1500 h data. Adding this 300 h to acoustic model training did not show any improvement on our test sets. Likewise, adding the corresponding transcripts to the LM did not improve the system. Therefore, we use this 300 h data as the DLM training data since it can be excluded from baseline system training without affecting the performance.
3) **Decode training data, generate lattices, extract N-best hypotheses.** We chose to use N=50 because prior experiments showed no significant improvements with larger N.
4) **Extract features.** We chose to use morph unigrams, bigrams, and trigrams as features in the DLM. The morphs are created from a simple Arabic segmentation process. Each vocabulary word is decomposed into morphs using context independent segmentation. Out of 795K words, 245K morphs were obtained.
5) **Word error information.** For each N-best sentence hypothesis, we compute the number of errors with respect to the training reference transcript.

To apply the DLM during testing, we also have to perform steps 3 and 4, i.e. decode the test data, generate N-best hypotheses, and extract features. The DLM then re-ranks the N-best hypotheses based on the extracted features.

The DLMs are first tested on a matched decoding setup, where the test data is decoded using the same acoustic and language models used for training, i.e. the P3U system. On the other hand, what we really wanted was to improve the performance when using our very best acoustic models trained for the Phase 5 DARPA GALE Evaluation. We therefore also test the DLMs on systems not matched to the DLM training.

In particular, we test on our two best Phase 5 Evaluation systems, trained on the full 1800 h of data and included sophisticated modeling and combinations, described in [29]. The two systems are

- **M+NNM** The M+NNM system is a multi-stream combination of two acoustic models M and NNM, where during decoding, the acoustic scores are computed as a weighted sum of scores from the two models. M and NNM both use a pronunciation dictionary (distinctly different from P3U) that is derived from MADA analysis, with short vowel information from predicted diacritics, plus pronunciation mapping rules. M uses conventional PLP front end while NNM uses neural network acoustic features.
- **U+BS+NNU** The U+BS+NNU system is a combination of three acoustic models: U, BS, and NNU. All three acoustic models use the un-vowelized grapheme dictionary similar to P3U. U is similar to P3U but has some tuning improvements in discriminative training and uses 300 h more training data. BS is a Bayesian Sensing acoustic model [18], [19], [20] and NNU is similar to U but has a neural network front end.

Another mismatch in testing the Phase 5 Evaluation systems is that the baseline language models are significantly improved. The LM is a linear interpolation of an un-pruned 916M 4-gram LM (created from 21 corpora), model M LMs [23] (trained on 7 important corpora), and a NNLM [21], [22] (trained on 3 important corpora).

We use two data sets for developmental tuning: dev07 (2.5 h) and dev08 (3 h). We report results on several test sets: dev09 (2.8 h); eval08 (3 h) and eval09 (4.2 h), the unsequestered portions of the GALE Phase 3 and 4 evaluation sets; and eval11 (3 h), the GALE Phase 5 evaluation set.

VI. Results

A. Supervised DLM Training

Our first experiment uses the perceptron algorithm for DLM training. We trained many DLMs by using α_0 values from 0 to 30, and the DLM with the best results on P3U dev07 was chosen. From all possible morph unigram, bigram, and trigram features, the perceptron algorithm selected a relatively small set of important features, about 1M features. Table I shows the perceptron DLM results when tested with a matched acoustic model (P3U). The improvements range from 0.2-0.6%.

TABLE I
Perceptron DLM results on matched system

	dev07	eval08	dev09	eval09
4-gram	10.5%	10.2%	15.6%	11.6%
Perceptron (P3U)	10.3%	9.9%	15.0%	11.4%

Then we performed experiments under mismatched training and test conditions. The trained DLMs remained the same but

the dev and test set N-best lists were generated with M+NNM and U+BS+NNU systems. There is considerable mismatch in such a test scenario since the acoustic model, pronunciation dictionary, and language model can all be different. For example, M+NNM system uses a different pronunciation dictionary (that includes short vowels and other diacritics not used by P3U). The acoustic model training data included the 300 h that was used for DLM training, and it has overall a much lower WER. Furthermore, before applying the DLM, the lattices were rescored with model M and word NNLM, resulting in a much stronger baseline. Can the DLM improve the test performance of the best eval systems even though the DLM training data was not prepared using those systems?

Table II shows the DLM results on the M+NNM system. The perceptron result uses the perceptron DLM (α_0 and iteration number) that optimizes dev07 on M+NNM. WER improves by 0.1% for the dev set and by 0.2% for the test sets. Where dev09 improved by 0.6% for the matched P3U system, the improvement is only 0.2% for M+NNM. This could be a result of mismatch in DLM training and testing, or simply because the baseline WER is much lower. It is nevertheless encouraging to see that improvements can be obtained even when the DLM is applied to a different system from the one it was trained on. Starting with the features selected by this perceptron DLM, we trained DLMs using GCLM or MBR.

For GCLM, we used a batch training implementation. Feature parameters were initialized with the perceptron DLM after scaling all the parameters by α_0 of the model. α_0 started as 1.0 but was updated during training. Tuning parameters included the regularization parameter σ and the step size ϵ. When properly regularized, the model converges nicely and it was not necessary to do early stopping based on a dev set. We ran the training for 500 iterations/epochs. dev07 was used to pick the best σ and ϵ; typical values that work well are $\sigma = 0.2$ and $\epsilon = 0.01$.

For MBR, we used an online training implementation without perceptron-style averaging. Tuning parameters included α_0, which we varied between 1 and 30, like for perceptron, and an initial step size ϵ_0. Typical values that work well are $\alpha_0 = 7$ and an initial step size of $\epsilon_0 = 0.1\alpha_0$. Intuitively, α_0 should not be too large; otherwise most of the posterior probability will be on the 1-best hypothesis and the update will be very little. We used the training set and dev07 to adjust the step size, using the heuristic to halve the step size whenever the training or dev set objective function or WER does not improve. We found that the model converges fairly quickly using this online implementation, sometimes under 20 epochs.

Table II also shows the results of DLMs trained with GCLM or MBR. The GCLM DLM is better than the perceptron DLM by 0.1% on dev09 and eval11. The MBR DLM further outperforms the GCLM DLM by 0.1% on all the test sets. Overall, on top of a very good baseline (model M + word NNLM), the MBR DLM achieves 0.1-0.4% improvement in WER, and the impact on eval11 is 0.4% absolute WER.

Table III shows the results on the U+BS+NNU system, also a mismatched testing condition. For this system, the performance of GCLM and MBR DLMs was very similar and was 0.1% better than the perceptron DLM for eval11. Again on top of a good baseline (model M + word NNLM), we obtain 0.2-0.3% improvement, specifically an absolute WER reduction of 0.3% on eval11.

TABLE II
DLM RESULTS ON MISMATCHED M+NNM SYSTEM

	dev07	dev09	eval09	eval11
4-gram	8.1%	13.3%	9.8%	9.3%
+ model M + word NNLM	7.5%	12.2%	9.1%	**8.5%**
Perceptron (M+NNM)	7.4%	12.0%	8.9%	8.3%
GCLM	7.5%	11.9%	8.9%	8.2%
MBR	7.4%	11.8%	8.8%	**8.1%**

TABLE III
DLM RESULTS ON MISMATCHED U+BS+NNU SYSTEM

	dev07	dev09	eval09	eval11
4-gram	8.2%	12.6%	9.5%	8.9%
+ model M + word NNLM	7.5%	11.7%	8.7%	**8.2%**
Perceptron (M+NNM)	7.4%	11.5%	8.6%	8.0%
GCLM	7.3%	11.6%	8.5%	7.9%
MBR	7.3%	11.5%	8.5%	**7.9%**

B. Unsupervised DLM Training

After the DARPA Evaluation, we performed some preliminary experiments to address the interesting scenario of training DLMs without transcripts. Table IV shows the results of unsupervised DLMs on the P3U system. The unsupervised perceptron DLM did not improve over baseline. However, with GCLM and MBR training, we obtained 0.1-0.2% absolute improvement in WER.

For GCLM training, we had to use a different dev set dev08 to do early stopping. Unlike supervised training, it was possible to over-train and degrade test performance, even with regularization. It was also necessary to use dev08 instead of dev07 for this purpose, probably because dev07 was used to select the features and this bias made it an unreliable dev set. For MBR training, dev07 was used for step size modification (described earlier) and dev08 was used to select the α_0. Interestingly, the optimal value for unsupervised training always turns out to be $\alpha_0 = 2$, compared with $\alpha_0 = 7$ for supervised training. This has the effect of spreading the posterior probabilities to more of the N-best hypotheses (rather than concentrating the probabilities in the top hypotheses.)

TABLE IV
UNSUPERVISED DLM RESULTS FOR P3U

	dev07	eval08	dev09	eval09
Baseline	10.5%	10.2%	15.6%	11.6%
Perceptron	10.4%	10.2%	15.6%	11.6%
GCLM	10.4%	10.1%	15.5%	11.5%
MBR	10.4%	10.0%	15.5%	11.5%

Conceptually, one big difference between perceptron and MBR DLM training is that perceptron training only makes updates if the 1-best hypothesis is different from the oracle hypothesis. It turns out that when the MBR hypothesis is used as the oracle in the training data, the oracle differs from the 1-best in only about 3,000 out of 160k sentences. MBR and

GCLM training have the distinct advantage that updates can happen even when the oracle and 1-best are the same.

TABLE V
UNSUPERVISED DLM RESULTS FOR M+NNM AND U+BS+NNU

	dev07	dev09	eval09	eval11
M+NNM System				
Baseline	7.5%	12.2%	9.1%	**8.5%**
Perceptron	7.6%	12.2%	9.0%	8.5%
GCLM	7.5%	12.2%	9.0%	8.4%
MBR	7.5%	12.2%	9.0%	**8.3%**
U+BS+NNU System				
Baseline	7.5%	11.7%	8.7%	**8.2%**
Perceptron	7.5%	11.7%	8.7%	8.2%
GCLM	7.5%	11.7%	8.6%	8.1%
MBR	7.5%	11.7%	8.6%	**8.1%**

Table V shows the results of unsupervised DLMs on the Phase 5 Evaluation systems. It is encouraging to see an improvement of 0.2% (from 8.5% to 8.3%) on the eval11 test set for the M+NNM system. This represents about half of the improvement we had obtained with the supervised DLM.

VII. CONCLUSION

In this paper we used discriminative language models to improve our best Arabic speech recognition systems for the Phase 5 DARPA GALE Evaluation. We explored different parameter estimation methods, including the minimum Bayes risk criterion, which minimizes the expected loss, calculated using the word error information and posterior probabilities of the N-best hypotheses of all the training sentences. One advantage of the MBR framework is that it is not necessary to designate a single reference or oracle hypothesis; this is useful since multiple hypotheses may turn out to have the same WER after text normalization (particularly relevant for Arabic).

During the Evaluation, a single MBR DLM thus trained was used to rescore the N-best hypotheses from all the systems with different acoustic models. Although there is significant mismatch between training and test conditions (different acoustic and language models), improvements were still observed. Finally we investigated unsupervised DLM training and found that in certain experiments, about half of the gain of the supervised DLM can be achieved. The DLM training data (300 h, 2.5M words) is much less than that for the baseline system (1800 h audio, 1.6B words for LM), so the gains are encouraging. Training on large amounts of untranscribed speech data may yield more improvements.

ACKNOWLEDGMENT

We would like to acknowledge the support of DARPA under Grant HR0011-06-2-0001 for funding part of this work. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense. We also thank the reviewers.

REFERENCES

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

[2] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," in *Proc. EMNLP*, Philadelphia, PA, USA, 2002, pp. 1 – 8.

[3] T. Oba, T. Hori, and A. Nakamura, "Round-robin discrimination model for reranking ASR hypotheses," in *Proc. Interspeech*, Makuhari, Japan, 2010, pp. 2446 – 2449.

[4] ——, "A comparative study on methods of weighted language model training for reranking LVCSR n-best hypotheses," in *Proc. ICASSP*, Dallas, Texas, USA, 2010, pp. 2446 – 2449.

[5] H.-K. J. Kuo, E. Fosler-Lussier, H. Jiang, and C.-H. Lee, "Discriminative training of language models for speech recognition," in *Proc. ICASSP*, Orlando, Florida, USA, 2002, pp. 325 – 328.

[6] A. Rastrow, A. Sethy, and B. Ramabhadran, "Constrained discriminative training of n-gram language models," in *Proc. ASRU*, Merano, Italy, 2009, pp. 311 – 316.

[7] M. Collins, M. Saraçlar, and B. Roark, "Discriminative syntactic language modeling for speech recognition," in *Proc. ACL*, Ann Arbor, MI, USA, 2005, pp. 507 – 514.

[8] E. Arısoy, M. Saraçlar, B. Roark, and I. Shafran, "Syntactic and sub-lexical features for Turkish discriminative language models," in *Proc. ICASSP*, Dallas, Texas, USA, 2010, pp. 5538 – 5541.

[9] I. Shafran and K. Hall, "Corrective models for speech recognition of inflected languages," in *Proc. EMNLP*, Sydney, Australia, 2006, pp. 390 – 398.

[10] E. Arısoy, B. Roark, I. Shafran, and M. Saraçlar, "Discriminative n-gram language modeling for Turkish," in *Proc. Interspeech*, Brisbane, Australia, 2008, pp. 825 – 828.

[11] N. Singh-Miller and M. Collins, "Trigger-based language modeling using a loss-sensitive perceptron algorithm," in *Proc. ICASSP*, Honolulu, Hawaii, USA, 2007, pp. 25 – 28.

[12] E. Arısoy, B. Ramabhadran, and H.-K. J. Kuo, "Feature combination approaches for discriminative language models," in *Proc. Interspeech*, Florence, Italy, 2011.

[13] M. Lehr and I. Shafran, "Discriminatively estimated joint acoustic, duration and language model for speech recognition," in *Proc. ICASSP*, Dallas, Texas, USA, 2010, pp. 5542 – 5545.

[14] V. Goel and W. J. Byrne, "Minimum Bayes-risk automatic speech recognition," *Computer Speech and Language*, vol. 14, pp. 115 – 135, 2000.

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

[16] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in n-best list rescoring," in *Proc. Eurospeech*, Rhodes, Greece, 1997.

[17] A. Deoras, D. Filimonov, M. Harper, and F. Jelinek, "Model combination for speech recognition using empirical Bayes risk minimization," in *Proc. SLT*, 2010.

[18] G. Saon and J.-T. Chien, "Discriminative training for Bayesian sensing hidden Markov models," in *Proc. ICASSP*, 2011, pp. 5316–5319.

[19] ——, "Bayesian sensing hidden Markov models for speech recognition," in *Proc. ICASSP*, 2011, pp. 5056–5059.

[20] ——, "Some properties of Bayesian sensing hidden Markov models," in *Proc. ASRU*, 2011.

[21] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Reseach*, vol. 3, 2003.

[22] A. Emami and L. Mangu, "Empirical study of neural network language models for Arabic speech recognition," in *Proc. ASRU*, Kyoto, Japan, Dec. 2007, pp. 147–152.

[23] S. F. Chen, "Shrinking exponential language models," in *Proc. NAACL-HLT*, 2009.

[24] G. Kurata, N. Itoh, and M. Nishimura, "Training of error-corrective model for ASR without using audio data," in *Proc. ICASSP*, Prague, Czech Republic, 2011.

[25] P. Jyothi and E. Fosler-Lussier, "Discriminative language modeling using simulated ASR errors," in *Proc. Interspeech*, Makuhari, Japan, 2010.

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

[27] D. A. Smith and J. Eisner, "Minimum risk annealing for training log-linear models," in *Proc. COLING/ACL*, Sydney, 2006, pp. 787 – 794.

[28] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, Orlando, FL, USA, 2002, pp. 105 – 108.

[29] L. Mangu, et. al., "The IBM 2011 GALE Arabic speech transcription system," in *Proc. ASRU*, 2011.