The IBM 2011 GALE Arabic Speech Transcription System

Lidia Mangu*, Hong-Kwang Kuo*, Stephen Chu*, Brian Kingsbury*, George Saon*, Hagen Soltau* and Fadi Biadsy‡

*IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598
‡Department of Computer Science, Columbia University, New York
{mangu,hkuo,smchu,bedk,gsaon,hsoltau}@us.ibm.com, fadi@cs.columbia.edu

***Abstract*—We describe the Arabic broadcast transcription system fielded by IBM in the GALE Phase 5 machine translation evaluation. Key advances over our Phase 4 system include a new Bayesian Sensing HMM acoustic model; multistream neural network features; a MADA vowelized acoustic model; and the use of a variety of language model techniques with significant additive gains. These advances were instrumental in achieving a word error rate of 7.4% on the Phase 5 evaluation set, and an absolute improvement of 0.9% word error rate over our 2009 system on the unsequestered Phase 4 evaluation data.**

***Index Terms*—large vocabulary speech recognition**

I. Introduction

The goal of the DARPA Global Autonomous Language Exploitation (GALE) program is to make Arabic and Chinese speech and text accessible to monolingual English speakers. The GALE program has sponsored annual evaluations of machine translation systems in which speech transcription is the first step when dealing with broadcast material. In this paper we describe IBM's 2011 transcription system for Arabic broadcasts, which was fielded in the GALE Phase 5 machine translation evaluation. Like most systems fielded in competitive evaluations, our system relies upon multiple passes of decoding, acoustic model adaptation, language model rescoring, and system combination to achieve the lowest possible word error rate. Because the 2011 system is similar to our previous evaluation system [1], we devote most of this paper to the novel aspects of the 2011 system. In Section II we describe the acoustic models used in our system, with an emphasis on the three new ones: Bayesian Sensing HMM acoustic model; acoustic models which use multistream neural network features; and a MADA vowelized acoustic model. Section III describes the various language models used for decoding, lattice or n-best rescoring, including an enhanced classing Model M, word and syntactic neural network, and discriminative language models. The system combination strategy and the overall system architecture are described in Section IV and Section V. We conclude in Section VI.

II. Acoustic models

We use an acoustic training set composed of approximately 1800 hours of transcribed Arabic broadcasts provided by the Linguistic Data Consortium (LDC) for the GALE evaluations.

Unless otherwise specified, all our acoustic models use 40-dimensional features that are computed by an LDA projection of a supervector composed from 9 successive frames of 13-dimensional mean- and variance-normalized PLP features followed by diagonalization using a global semi-tied covariance transform [2], use pentaphone cross-word context with a "virtual" word-boundary phone symbol that occupies a position in the context description, but does not generate an acoustic observation. Speaker-adapted systems are trained using VTLN and fMLLR. All the models use variable frame rate processing [3].

Given that the short vowels and other diacritic markers are typically not orthographically represented in Arabic texts, we have a number of choices for building pronunciation dictionaries: 1) Unvowelized (graphemic) dictionaries in which the short vowels and diacritics are ignored 2) Vowelized dictionaries which use the Buckwalter morphological analyzer [4] for generating possible vowelized pronunciations and 3) Vowelized dictionary which uses the output of a morphological analysis and disambiguation tool (MADA) [5]; the assignment of such diacritic markers is based on the textual context of each word (to distinguish word senses and grammatical functions). Our 2011 transcription system uses the acoustic models described below.

- **SI** A speaker-independent, unvowelized acoustic model trained using model-space boosted maximum mutual information [6]. The PLP features for this system are only mean-normalized. The **SI** model comprises 3K states and 151K Gaussians.
- **U** A speaker-adapted, unvowelized acoustic model trained using both feature- and model-space BMMI. The **U** model comprises 5K states and 803K Gaussians.
- **SGMM** A speaker-adapted, Buckwalter vowelized subspace Gaussian mixture model [7], [8] trained with feature- and model-space versions of a discriminative criterion based on both the minimum phone error (MPE) [9] and BMMI criteria. The **SGMM** model comprises 6K states and 150M Gaussians that are represented using an efficient subspace tying scheme.
- **V** A speaker-adapted, Buckwalter vowelized acoustic model trained using the feature-space BMMI and model-space MPE criteria. The changes in this model compared

to all the other models: 1) the "virtual" word boundary phones are replaced with word-begin and word-end tags 2) it uses a dual decision tree that specifies 10K different Gaussian mixture models, but 50K context-dependent states, 3) it uses a single, global decision tree and 4) expands the number of phones on which a state can be conditioned to ± 3 within words. This model has 801K Gaussians.

- **BS** A speaker-adapted, unvowelized acoustic model using Bayesian sensing HMMs where the acoustic feature vectors are modeled by a set of state-dependent basis vectors and by time-dependent sensing weights [10]. The Bayesian formulation comes from assuming state-dependent Gaussian priors for the weights and from using marginal likelihood functions obtained by integrating out the weights. The marginal likelihood is Gaussian with a factor analyzed covariance matrix with the basis providing a low-rank correction to the diagonal covariance of the reconstruction error [11]. The details of this model are given in Section II-A.
- **M** A speaker-adapted system, MADA vowelized system, with an architecture similar to **V**. The details of this model are given in Section II-B.
- **NNU, NNM** Speaker-adapted acoustic models which use neural network features. They were built using either the unvowelized lexicon (**NNU**) or the MADA one (**NNM**). Section II-C describes these models in more detail.

A. Bayesian Sensing HMMs (BS)

1) Model description: Here, we briefly describe the main concepts behind Bayesian sensing hidden Markov models [10]. The state-dependent generative model for the D-dimensional acoustic feature vectors $\mathbf{x}_t$ is assumed to be

$$\mathbf{x}_t = \Phi_i \mathbf{w}_t + \boldsymbol{\epsilon}_t \tag{1}$$

where $\Phi_i = [\boldsymbol{\phi}_{i1}, \ldots, \boldsymbol{\phi}_{iN}]$ is the basis (or dictionary) for state i and $\mathbf{w}_t = [w_{t1}, \ldots, w_{tN}]^T$ is a time-dependent weight vector. The following additional assumptions are made: (1) when conditioned on state i, the reconstruction error is zero-mean Gaussian distributed with precision matrix R_i, i.e. $\boldsymbol{\epsilon}_t | s_t = i \sim \mathcal{N}(\mathbf{0}, R_i^{-1})$ and (2) the state-conditional prior for $\mathbf{w}_t$ is also zero-mean Gaussian with precision matrix A_i, that is $\mathbf{w}_t | s_t = i \sim \mathcal{N}(\mathbf{0}, A_i^{-1})$. It can be shown that, under these assumptions, the marginal state likelihood $p(\mathbf{x}_t | s_t = i)$ is also zero-mean Gaussian with the factor analyzed covariance matrix [11]

$$S_i \triangleq R_i^{-1} + \Phi_i A_i^{-1} \Phi_i^T \tag{2}$$

In summary, the state-dependent distributions are fully characterized by the parameters $\{\Phi_i, R_i, A_i\}$. In [10], we discuss the estimation of these parameters according to a maximum likelihood type II criterion, whereas in [12] we derive parameter updates under a maximum mutual information objective function.

2) Automatic relevance determination: For diagonal $A_i = \mathrm{diag}(\alpha_{\mathrm{i1}}, \ldots, \alpha_{\mathrm{iN}})$, the estimated precision matrix values α_{ij} encode the relevance of the basis vectors $\boldsymbol{\phi}_{ij}$ for the dictionary representation of $\mathbf{x}_t$. This means that one can use the trained α_{ij} for controlling model complexity. One can first train a large model and then prune it to a smaller size by discarding the basis vectors which correspond to the largest precision values of the sensing weights.

3) Initialization and training: We first train a large acoustic model with 5000 context-dependent HMM states and 2.8M diagonal covariance Gaussians using maximum likelihood in a discriminative FMMI feature space. The means of the GMM for each state are then clustered using k-means. The initial bases are formed by the clustered means. The resulting number of mixture components for the Bayesian sensing models after the clustering step was 417K. The precision matrices for the sensing weights and the reconstruction errors are assumed to be diagonal and are initialized to the identity matrix.

The models are trained with 6 iterations of maximum likelihood type II estimation. Next, we discard 50% of the basis vectors corresponding to the largest precision values of the sensing weights and retrain the pruned model for two additional ML type II iterations. We then generate numerator and denominator lattices with the pruned models and perform 4 iterations of boosted MMI training of the model parameters as described in [12]. The effect of pruning and discriminative training is discussed in more details in [11].

B. MADA-based acoustic model (M)

This acoustic model is similar to the **V** model. It uses a global tree, word position tags, and a large phonetic context of ± 3 . While the MADA-based model uses approximately the same number of Gaussians, the decision tree uses only one level, keeping the number of HMM states to $10,000$. Since the MADA-based model uses a smaller phone set than the Buckwalter vowelized models, we were able to reuse the Vowelized alignments and avoid the flatstart procedure. In this section we describe the strategy used for constructing training and decoding pronunciation dictionaries, the main difference between this system and the **V** system. Both pronunciation dictionaries are generated following [13] with some slight modification.

1) Training Pronunciation Dictionary: Here we describe an automatic approach to building a pronunciation dictionary that covers all words in the orthographic transcripts of the training data. First, for each utterance transcript, we run MADA to disambiguate each word based on its context in the transcript. MADA outputs all possible fully-diacritized morphological analyses for each word, ranked by their confidence, the MADA confidence score. We thus obtain a fully-diacritized orthographic transcription for training. Second, we map the highest-ranked diacritization of each word to a set of pronunciations, which we obtain from the 15 pronunciation rules described in

[13]. Since MADA may not always rank the best analysis as its top choice, we also run the pronunciation rules on the **second** best choice returned by MADA, when the difference between the top two choices is less than a threshold determined empirically (in our implementation we chose 0.2). The IBM system is flexible enough to allow specifying multiple diacritized word options at the (training) transcript level. A sentence can be a sequence of fully diacritized word pairs as opposed to a sequence of single words. This whole process gives us fully disambiguated and diacritized training transcripts with more than one or two options per word.

2) Decoding Pronunciation Dictionary: For building the decoding dictionary we run MADA on the transcripts of the speech training data as well as on the Arabic Gigaword corpus. In this dictionary, all pronunciations produced (by the pronunciation rules) for all diacritized word instances (from MADA first and second choices) of the same undiacritized form are mapped to the undiacritized and *normalized* word form. Word normalization here refers to removing diacritic markers and replace Buckwalter normalized Hamzat-Wasl ($\{$), $<$, and $>$ by the letter 'A'. Note that it is standard to produce undiacritized transcripts when recognizing MSA. Diacritization is generally not necessary to make the transcript readable by Arabic-literate readers. Therefore, entries in the decoding pronunciation dictionary need only to consist of undiacritized words mapped to a set of phonetically-represented diacritizations.

A pronunciation confidence score is calculated for each pronunciation. We compute a pronunciation score s for a pronunciation p as the average of the MADA confidence scores of the MADA analyses of the word instances that this pronunciation was generated from. We compute this score for each pronunciation of a normalized undiacritized word. Let m be the maximum of these scores. Now, the final pronunciation confidence score for p is $-log_{10}(c/m)$. This basically means that the best pronunciation receives a penalty of 0 when chosen by the ASR decoder. This dictionary has about 3.6 pronunciations per word when using the first and second MADA choices.

*C. Neural Network acoustic models (***NNU** *and* **NNM***)*

The neural network feature extraction module uses two feature streams computed from mean and variance normalized, VTLN log Mel spectrograms, and is trained in a piecewise manner, in which (1) a state posterior estimator is trained for each stream, (2) the unnormalized log-posteriors from all streams are summed together to combine the streams, and (3) features for recognition are computed from the bottleneck layer of an autoencoder network. One stream, the lowpass stream, is computed by filtering the spectrograms with a temporal lowpass filter, while the other stream, the bandpass stream, is computed by filtering the spectrograms with a temporal bandpass filter. Both filters are 19-point FIR filters. The lowpass filter has a cutoff frequency of 24 Hz. The bandpass filter has a differentiator-like (gain proportional to frequency) response from 0–16 Hz and a high-pass cutoff frequency of 27 Hz. The posterior estimators for each stream

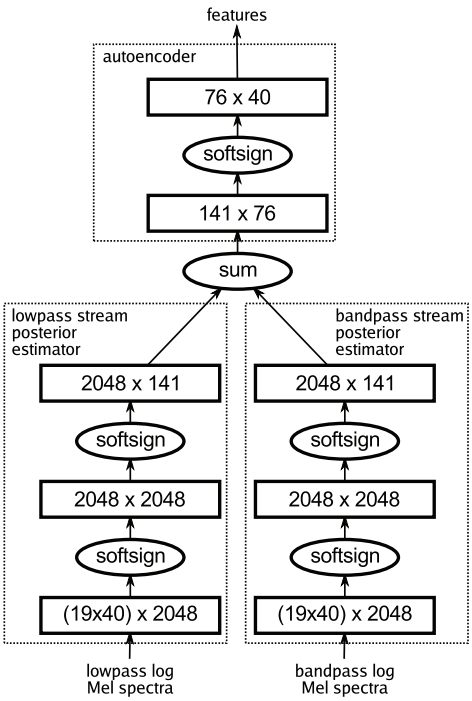

Fig. 1. Structure of the multistream, discriminatively trained neural network feature extraction module. The dotted boxes enclose modules that are trained separately.

compute the probabilities of 141 context-independent HMM states given an acoustic input composed from 19 frames of 40-dimensional, filtered spectrograms. They have two 2048-unit hidden layers, use softsign nonlinearities [14] between layers, and use a softmax nonlinearity at the output. The softsign nonlinearity is $y = x/(1 + |x|)$. Initial training optimizes the frame-level cross-entropy criterion. After convergence, the estimators are further refined to discriminate between state sequences using the minimum phone error criterion [15], [16]. Stream combination is performed by discarding the softmax output layer for each stream posterior estimator, and summing the resulting outputs, which may be interpreted as unnormalized log-posterior probabilities. We then train another neural network, containing a 40-dimensional bottleneck layer, as an autoencoder, and use the trained network to reduce the dimensionality of the neural network features. The original autoencoder network has a first hidden layer of 76 units, a second hidden layer of 40 units, a linear output layer, and uses softsign nonlinearities. The training criterion for the autoencoder is the cross-entropy between the *normalized* posteriors generated by processing the autoencoder input and output vectors through a softmax nonlinearity. Once the autoencoder is trained, the second layer of softsign nonlinearities and the weights that expand from the 40-dimensional bottleneck layer back to the 141-dimensional output are removed. The overall struture of the neural network feature extraction module is illustrated in Figure 1.

Once the features are computed, the remaining acoustic

modeling steps are conventional, using 600K 40-dimensional Gaussians modeling 10K quinphone context-dependent states, where we do both feature- and model-space discriminative training using the BMMI criterion. Two acoustic models were trained using the neural-net features: one (**NNM**) used a MADA-vowelized lexicon, while the other (**NNU**) used an unvowelized lexicon. Note that the posterior estimators used in feature extraction were trained with MADA-vowelized alignments.

III. Language models

For training language models we use a collection of 1.6 billion words, which we divide into 20 different sources. The two most important components are the broadcast news (**BN**) and broadcast conversation (**BC**) acoustic transcripts (7.5 million words each) corresponding to 1800h of speech transcribed by LDC for the GALE program. Other notable sources: Arabic Gigaword corpus, 29M words of transcripts harvested from the web (**Archive**), Arabic text from parallel corpora used for machine translation, etc. We use a vocabulary of 795K words, which is based on all available corpora, and is designed to completely cover the acoustic transcripts. To build the baseline language model, we train a 4-gram model with modified Kneser-Ney smoothing [17] for each source, and then linearly interpolate the 20 component models with the interpolation weights chosen to optimize perplexity on a held-out set. We combine all the 20 components into one language model using entropy pruning [18]. By varying the pruning thresholds we create 1) a 913M n-gram LM (no pruning) to be used for lattice rescoring (**Base**) and 2) a 7M n-gram LM to be used for the construction of static, finite-state decoding graphs.

In addition to the baseline language models described above, we investigated various other techniques which differ in either the features they employ or the modeling strategy they use. These are described below.

- **ModelM** A class-based exponential model [20]. Compared to the models used in the previous evaluation, we use a new enhanced word classing [21]. The bigram mutual information clustering method used to derive word classes in the original Model M framework is less than optimal due to mismatches between the classing objective function and the actual LM, so the new method attempts to address this discrepancy. Key features of the new method include: a) a class-based model that includes word n-gram features to better mimic the nature of the actual language modeling, b) a novel technique for estimating the likelihood of unseen data for the clustering model, and c) n-gram clustering compared to bigram clustering in the original method. We build Model M models with improved classing on 7 of the corpora with the highest interpolation weights in the baseline model.
- **WordNN** A 6-gram neural network language model using word features; Compared to the model used in the P4 evaluation [1], we train on more data (44 milion words of data from **BN**, **BC** and **Archive**). We also enlarge the neural network architecture (increased the feature vector dimension from 30 to 120 and the number of hidden units from 100 to 800) and normalize the models. We create a new LM for lattice rescoring by interpolating this model with the 7 **ModelM** models and **Base**, with the interpolation weights optimized on the held-out set. In the previous evaluation we did not get an improvement by interpolating the wordNN model with model M models, but the changes made this year result in significant improvements.
- **SyntaxNN** A neural network language model using syntactic and morphological features [19]. The syntactic features include exposed head words and their non-terminal labels, both before and after the predicted word. For this neural network model we used the same training data and the same neural network architecture as the one described for **WordNN**. This language model is used for n-best rescoring.
- **DLM** A discriminative language model trained using the Minimum Bayes Risk (MBR) criterion [22]. Unlike the other LMs, a DLM is trained on patterns of confusion or errors made by the speech recognizer. Our potential features consist of unigram, bigram, and trigram morphs, and we used the perceptron algorithm to select a small set of useful features. With the selected features, we trained the DLM using an MBR-based algorithm, which minimizes the expected loss, calculated using the word error information and posterior probabilities of the N-best hypotheses of all the training sentences. To prepare data for DLM training, we used a Phase 3 unvowelized recognizer trained on 1500 hr. of acoustic data to decode an un-seen 300 hr. set. This un-seen training set was provided in Phase 4, but adding this data to acoustic or language model training did not improve the system, so it is an ideal set for DLM training. During the evaluation, a single MBR DLM thus trained was used to rescore the N-best hypotheses from all the systems. Although there is a mismatch between training and test conditions, improvements were still observed. Details of these experiments as well as post-eval experiments are presented in [22].

Having many diverse language models, the challenge is to be able to combine them while achieving additive gains, and Section V describes our strategy.

IV. System combination

We employ three different techniques for system combination. The first technique is cross-adaptation ($\times$), where the fMLLR and MLLR transforms required by a speaker-adapted acoustic model are computed using transcripts from some other, different speaker-adapted acoustic model. The second technique is tree-array combination (**+**), a form of multi-stream acoustic modeling in which the acoustic scores are computed as a weighted sum of scores from two or more models that can have different decision trees [23]. The only requirement for the tree-array combination is that the individual models are built

using the same pronunciation dictionary. The third technique is hypothesis combination using the `nbest-rover` [24] tool from the SRILM toolkit [25]. In all these combination strategies, the choice of systems to combine was based on performance on a variety of development sets.

V. System Architecture

The IBM's 2011 GALE Arabic transcription system is a sequence of multiple passes of decoding, acoustic model adaptation, language model rescoring, and system combination steps. In this section, we show how all the models described in the previous sections are combined to generate the final transcripts. We report results on several data sets: DEV'07 (2.5 hours); DEV'09 (2.8 hours); EVAL'09, the unsequestered portion of the GALE Phase 4 evaluation set (4.2 hours); and EVAL'11, the GALE Phase 5 evaluation set (3 hours). EVAL'11 is unseen data on which no tuning was done. In our 2011 evaluation system we have the following steps.

1) Cluster the audio segments into hypothesized speakers.
2) Decode with the **SI** model.
3) Compute VTLN warp factors per speaker using transcripts from (2)
4) Decode using the **U** model cross-adapted on **SI**
5) Decode using the **SGMM** model cross-adapted on (4)
6) Compute best frame rates per utterance using transcripts from (5),
7) Decode using the **SGMM** model cross-adapted on **U** and frame rates from (6)
8) a) Using the **U** model and transcripts from (5), compute fMLLR and MLLR transforms.
   b) Using the **BS** model and transcripts from (5), compute fMLLR and MLLR transforms.
   c) Using the **NNU** model and transcripts from (5), compute fMLLR and MLLR transforms.
   d) Decode and produce lattices using a tree-array combination of the **U** model with transforms from (8a), the **BS** model with transforms from (8b) and **NNU** with transforms from (8c)
9) a) Using the **M** model and transcripts from (5), compute fMLLR and MLLR transforms.
   b) Using the **NNM** model and transcripts from (5), compute fMLLR and MLLR transforms.
   c) Decode and produce lattices using a tree-array combination of the **M** model with transforms from (9a) and the **NNM** model with transforms from (9b)
10) Decode and produce lattices using the **V** model, frame rates from (6) and fMLLR and MLLR transforms computed using transcripts from (5).
11) Using an interpolation of **Base**, 7 **ModelM** and one **WordNN** language models
   a) rescore lattices from (8d), extract 50-best hypotheses
   b) rescore lattices from (9c), extract 50-best hypotheses
   c) rescore lattices from (10), extract 50-best hypotheses
12) Parse the 50-best lists from (11) and score them with a **SyntaxNN** language model; produce new language model scores for each hypothesis
13) Score the 50-best lists from (11) with a discriminative language model and produce new language model scores for each hypothesis.
14) Combine acoustic scores, language model scores from (11), syntax LM scores from 12) and discriminative LM scores from 13 using simplex
   a) add the new scores to the hypotheses from (11a)
   b) add the new scores to the hypotheses from (11b)
   c) add the new scores to the hypotheses from (11c)
15) Combine the hypotheses from (14a), (14b), and (14c) using the `nbest-rover` tool from the SRILM toolkit [25]. This is the final output.

Step	Decoding Pass	DEV'09	EVAL'09	EVAL'11
(8d)	(U+BS+NNU)xSGMMxU	12.6%	9.5%	8.9%
(9c)	(M+NNM)xSGMMxU	13.3%	9.8%	9.2%
(10)	VxSGMMxU	13.5%	9.8%	9.5%
(14a)	(8d) + simplex	11.4%	8.4%	7.8%
(14b)	(9c) + simplex	11.9%	8.6%	8,1%
(14c)	(10) + simplex	12.5%	9.2%	8.4%
(15)	(14a) + (14b) + (14c)	11.1%	8.1%	7.4%

TABLE I
Word error rates for the final three combined models before and after adding LM rescoring passes

Step	Language Model	DEV'09	EVAL'09	EVAL'11
(8d)	Base	12.6%	9.5%	8.9%
(11)	+ ModelM and WordNN	11.7%	8.8%	8.2%
(12)	+ SyntaxNN	11.6%	8.6%	7.9%
(13)	+ DLM	11.5%	8.6%	8.0%
(14)	+ SyntaxNN and DLM	11.4%	8.4%	7.8%

TABLE II
Word error rates for different LM rescoring steps on the (U+BS+NNU)×SGMM×U.vfr (8d) set of lattices

Table II shows the word error rates obtained after adding new language models either for lattice or n-best rescoring for the (8d) system. It can be seen that each additional rescoring pass improves the performance, and that the total improvement from language modeling rescoring is 1.1-1.2% absolute on all the sets. Similar improvements have been obtained on the other two systems that are part of the final system combination ((9c) and (10)).

A. Simplified Architecture

After the P5 evaluation we investigated an alternative simpler system architecture in which we eliminate the tree-array combination and cross-adaptation steps. In the new simplified system we combine the 50-best hypotheses from **U**, **BS**, **V**, **M**, **NNM** and **SGMM** using n-best rover. For this comparison experiment we use the language model from (11). In order to have a fair comparison we redid all the steps in the P5

Architecture	DEV'07	DEV'09	EVAL'09
Eval P5	7.1%	11.5%	8.5%
Simplified	7.2%	11.5%	8.5%

TABLE III
COMPARISON OF THE SYSTEM ARCHITECTURE USED IN THE P5 EVALUATION AND THE SIMPLIFIED ONE

evaluation using this language model. Table III shows that almost the same results are obtained in both cases, suggesting that the gains are coming from system diversity not from the combination method. Also, by comparing the Eval P5 final WER number with the one from the actual evaluation we see that it was worth using more expensive processing for LM (i.e. generating N-bests, using syntax LM, DLM); it actually helps by 0.4% to the bottom line performance.

VI. SUMMARY

In this paper we present IBM's 2011 GALE Arabic speech transcription system, describing improvements made over the past year that led to a word error rate of 7.4% on the 2011 evaluation data and a year-to-year, absolute reduction of 0.9% word error rate on the unsequestered 2009 evaluation data. New techniques that contributed to this improvement include Bayesian Sensing HMM acoustic models, improved neural network acoustic features, MADA vowelized acoustic model, improved word and syntax neural network language models and enhanced classing Model M and discriminative language models.

ACKNOWLEDGMENT

This work was supported in part by DARPA under Grant HR0011-06-2-0001[1].

REFERENCES

[1] B. Kingsbury, H. Soltau, G. Saon, S. Chu, H.-K. Kuo, L. Mangu, S. Ravuri, N. Morgan, and A. Janin, "The IBM 2009 GALE Arabic speech transcription system," in *Proc. ICASSP*, 2011, pp. 4378–4381.

[2] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[3] S. M. Chu and D. Povey, "Speaking rate adaptation using continuous frame rate normalization," in *Proc. ICASSP*, 2010, pp. 4306–4309.

[4] T. Buckwalter, "LDC2004L02: Buckwalter Arabic morphological analyzer version 2.0," 2004, Linguistic Data Consortium.

[5] Nizar Habash and Owen Rambow, "Arabic Diacritization through Full Morphological Tagging," in *Proceedings of the 8th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL07)*, 2007.

[6] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. ICASSP*, 2008, vol. II., pp. 4057–4060.

[7] D. Povey *et al.*, "Subspace Gaussian mixture models for speech recognition," in *Proc. ICASSP*, 2010, pp. 4330–4333.

[8] G. Saon, H. Soltau, U. Chaudhari, S. Chu, B. Kingsbury, H.-K. Kuo, L. Mangu, and D. Povey, "The IBM 2008 GALE Arabic speech transcription system," in *Proc. ICASSP*, 2010, pp. 4378–4381.

[1]Approved for Public Release, Distribution Unlimited. 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.

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

[10] G. Saon and J.-T. Chien, "Bayesian sensing hidden Markov models for speech recognition," in *Proc. of ICASSP*, 2011, pp. 5056–5059.

[11] G. Saon and J.-T. Chien, "Some properties of Bayesian sensing hidden Markov models," in *Proc. of IEEE ASRU*, 2011, Submitted.

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

[13] F. Biadsy, N. Habash, and J. Hirschberg, "Improving the Arabic Pronunciation Dictionary for Phone and Word Recognition with Linguistically-Based Pronunciation Rules," in *Proceedings of NAACL/HLT 2009, Colorado, USA*, 2009.

[14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[15] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, 2002.

[16] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. ICASSP*, 2009.

[17] S. F. Chen and J. T. Goodman, "An empirical study of smoothing techniques for language modeling," Tech. Rep. TR-10-98, Harvard University, 1998.

[18] A. Stolcke, "Entropy-based pruning of backoff language models," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 270–274.

[19] H.-K. J. Kuo, L. Mangu, A. Emami, I. Zitouni, and Y.-S. Lee, "Syntactic features for Arabic speech recognition," in *Proc. ASRU*, 2009.

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

[21] S.F. Chen and S.M. Chu, "Enhanced word classing for model m," in *Proc. Interspeech*, 2010, pp. 1037–1040.

[22] H. Kuo, L. Mangu, E. Arisoy, and G. Saon, "Minimum Bayes Risk Discriminative Language Models for Arabic Speech Recognition," in *Proc. of IEEE ASRU*, 2011, Submitted.

[23] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEE Workshop on Spoken Language Technology*, 2010.

[24] A. Stolcke *et al.*, "The SRI March 2000 Hub-5 conversational speech transcription system," in *Proc. NIST Speech Transcription Workshop*, 2000.

[25] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. ICSLP*, 2002, pp. 901–904.