THE IBM 2009 GALE ARABIC SPEECH TRANSCRIPTION SYSTEM

Brian Kingsbury, Hagen Soltau, George Saon,
Stephen Chu, Hong-Kwang Kuo, Lidia Mangu

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA

Suman Ravuri, Adam Janin, Nelson Morgan

International Computer Science Institute
Berkeley, CA 94704, USA

ABSTRACT

We describe the Arabic broadcast transcription system fielded by IBM in the GALE Phase 4 machine translation evaluation. Key advances over our Phase 3.5 system include improvements to context-dependent modeling in vowelized Arabic acoustic models; the use of neural-network features provided by the International Computer Science Institute; Model M language models (a novel, class-based exponential model); a neural network language model that uses syntactic and morphological features; and improvements to our system combination strategy. These advances were instrumental in achieving a word error rate of 8.9% on the Phase 4 evaluation set, and an absolute improvement of 1.6% word error rate over our 2008 system on the unsequestered Phase 3.5 evaluation data.

Index Terms— large vocabulary speech recognition

1. INTRODUCTION

The purpose of the DARPA Global Autonomous Language Exploitation (GALE) program is to make Arabic and Chinese broadcasts, newswire, and web logs accessible to monolingual English speakers. To that end, the GALE program has sponsored annual competitive evaluations of machine translation systems in which speech transcription is a necessary front-end for broadcast material. Here we describe IBM's 2009 transcription system for Arabic broadcasts, which was fielded in the GALE Phase 4 machine translation evaluation. Key innovations in this year's system include improvements to context-dependent modeling in vowelized Arabic acoustic models; the use of neural-network features provided by the International Computer Science Institute (ICSI); Model M language models [1]; a neural network language model that uses syntactic and morphological features [2]; and improvements to our system combination strategy. These advances were instrumental in achieving a word error rate of 8.9% on the Phase 4 evaluation set, and an absolute improvement of 1.6% word error rate over our 2008 system on the unsequestered Phase 3.5 evaluation data.

2. OVERVIEW

Like other transcription systems fielded in competitive evaluations, IBM's 2009 GALE Arabic transcription system relies upon multiple passes of decoding, acoustic model adaptation, language model rescoring, and system combination to achieve the lowest possible word error rate. In this section, we briefly describe the components of the 2009 system and the processing steps used to produce the final transcripts. Because the 2009 system is similar to our 2008 system [3], we devote most of this paper to the novel aspects of the 2009 system.

2.1. Acoustic models

Our 2009 transcription system uses the five different acoustic models described below. Unless otherwise specified, all 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 perceptual linear prediction (PLP) [4] features followed by diagonalization using a global semi-tied covariance transform [5] and 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 vocal tract length normalization (VTLN) [6] and feature-space maximum-likelihood linear regression (fMLLR, also called constrained MLLR) [7].

SI A speaker-independent, unvowelized (graphemic) acoustic model trained using model-space boosted maximum mutual information [8]. 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. This model also made use of variable frame rate processing [9]. The **U** model comprises 5K states and 803K Gaussians.

SGMM A speaker-adapted, vowelized subspace Gaussian mixture model [10, 3] trained with feature- and model-space versions of a discriminative criterion based on both the minimum phone error (MPE) [11] and BMMI criteria. This model also made use of variable frame rate processing [9]. The **SGMM** model comprises 6K states and 150M Gaussians that are represented using an efficient subspace tying scheme.

V A speaker-adapted, vowelized (phonemic) acoustic model trained using the feature-space BMMI and model-space MPE criteria. This model differs from the others in its treatment of context and tying structure, the details of which are given in Section 3. The **V** model comprises 50K states and 801K Gaussians.

NN A speaker-adapted, vowelized acoustic model trained using the feature-space BMMI and model-space MPE criteria. This model differs from the others in that it uses neural-network features provided by the International Computer Science Institute. Section 4 describes these features in more detail. The **NN** model comprises 10K states and 889K 36-dimensional Gaussians.

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 Phase 4 evaluation and 85 hours of FBIS and TDT-4 data with transcripts provided by BBN.

We report results on several data sets: DEV'07 (2.5 hours); EVAL'08U, the unsequestered portion of the GALE Phase 3.5 evaluation set (3 hours); and EVAL'09, the GALE Phase 4 evaluation set (16 hours). DEV'07 and EVAL'08U are development data: our models were tuned on these sets. EVAL'09 is unseen data on which no tuning was done.[1]

2.2. Language models

The language model training data is a collection of about 1.6 billion words provided by the LDC. We divide the training corpus into 20 different sources, including acoustic transcripts (split into broadcast news and broadcast conversation genres), different portions of the Arabic Gigaword corpus (LDC2009T30), Arabic text from parallel corpora used for machine translation training, etc. To build the baseline language model, we train a 4-gram model with modified Kneser-Ney smoothing [12] for each source, and then linearly interpolate the 20 component models with the interpolation weights chosen to optimize perplexity on a held-out set. Typically, the language models corresponding to the audio transcripts (roughly 15 million words) have the highest weights because they are best matched to the domain of interest. The resulting interpolated language model is pruned using entropy pruning [13] to about 7M n-grams for the construction of static, finite-state decoding graphs. The unpruned model, which contains 883M n-grams, is used for lattice rescoring. We use a vocabulary of 795K words, which is based on all available corpora, and is designed to completely cover the acoustic transcripts.

In previous years [3] we interpolated a large unpruned language model with a word-based neural network language model (NNLM). This year we enriched our language models by adding Model M (a class-based exponential model) [1] and neural network language models using syntactic features [2]. These new models are described in Sections 5 and 6.

2.3. System combination

We employ three different techniques for system combination. The first technique is cross-adaptation, 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 a form of multi-stream acoustic modeling in which the acoustic scores (weighted negative log-likelihoods) are computed as a weighted sum of scores from two or more models that can have different decision trees [14]. The third technique is hypothesis combination using the `nbest-rover` [15] tool from the SRILM toolkit [16]. The choice of systems to combine and the weights the systems receive in the combination process was based on performance on the GALE DEV'07, DEV'08, DEV'09, and EVAL'08U sets.

2.4. System architecture

Our 2009 transcription system uses the following steps.

1. Cluster the audio segments into hypothesized speakers.
2. Decode with the **SI** model.
3. Using transcripts from (2), compute VTLN warp factors per speaker.
4. Using the **U** model and transcripts from (2), compute fMLLR and MLLR transforms, then decode.

Step	Decoding pass	DEV'07	EVAL'08U	EVAL'09
(2)	SI	16.7%	15.3%	16.6%
(4)	U	10.4%	9.9%	11.5%
(5)	SGMMxU	8.4%	8.7%	10.3%
(7)	SGMMxU.vfr	8.4%	8.6%	10.0%
(8)	UxSGMM.vfr	8.9%	8.7%	10.3%
(9c)	V_NNxSGMM	8.4%	8.4%	10.1%
(10)	Model M on (7)	7.9%	8.0%	9.6%
(11)	syntax on (10)	7.6%	7.6%	9.3%
(12)	Model M on (8)	8.3%	8.2%	9.7%
(13)	Model M on (9c)	8.1%	7.9%	9.6%
(14)	(11) + (12) + (13)	7.4%	7.3%	8.9%

Table 1. % word error rates for different stages in our 2009 GALE Arabic transcription system on DEV'07, EVAL'08U, and EVAL'09.

5. Using the **SGMM** model and transcripts from (4), compute fMLLR and MLLR transforms, then decode.
6. Using transcripts from (5), compute best frame rates per utterance.
7. Using the **SGMM** model, transcripts from (5), and frame rates from (6), compute fMLLR and MLLR transforms, then decode and produce lattices.
8. Using the **U** model, transcripts from (7), and frame rates from (6), compute fMLLR and MLLR transforms, then decode and produce lattices.
9. (a) Using the **V** model and transcripts from (5), compute fMLLR and MLLR transforms.

   (b) Using the **NN** model and transcripts from (5), compute fMLLR and MLLR transforms.

   (c) Using the **V** model with transforms from (9a) and the **NN** model with transforms from (9b) together as a multi-stream acoustic model, decode and produce lattices.
10. Rescore the lattices from (7) using an interpolation of nine Model M language models, and extract the 200-best hypotheses for each utterance.
11. Score the 200-best lists from (10) with a neural network language model that uses syntax features, and produce new language model scores that interpolate the Model M and syntax language model scores.
12. Rescore the lattices from (8) using an interpolation of nine Model M language models, and extract the 100-best hypotheses for each utterance.
13. Rescore the lattices from (7) using an interpolation of nine Model M language models, and extract the 100-best hypotheses for each utterance.
14. Combine the hypotheses from (11), (12), and (13) using the `nbest-rover` tool from the SRILM toolkit [16].

3. CONTEXT MODELING FOR VOWELIZED ARABIC

One way to exploit a large amount of training data, as we have for the GALE Arabic task, is to build very detailed acoustic models by extending the size of the context. For the **V** models we made four changes to the context modeling that proved to be beneficial.

[1] Note that under GALE program rules, only the unsequestered portion of EVAL'09 can be used for future system development.

word boundary marker	within-word context	global tree?	DEV'07
virtual phone	2	no	13.3%
wb & we tags	2	no	13.1%
wb & we tags	3	no	13.0%
wb & we tags	3	yes	12.9%

Table 2. % word error rates on DEV'07 for improved forms of context modeling for vowelized Arabic acoustic models.

tree	DEV'07
standard	12.8%
dual	12.3%

Table 3. % word error rates on DEV'07 for a standard decision tree specifying 10K GMMs and a dual tree that specifying 50K states sharing 10K GMMs.

1. Instead of marking word boundaries with a "virtual" word boundary phone that occupies a position in the context description, but does not generate an acoustic observation, we used word-begin and word-end tags to label phones in the start-of-word and end-of word positions.
2. We expanded the number of phones on which a state can be conditioned to ±3 within words, while keeping the extent of cross-word context dependency to one phone.
3. We used a single, global decision tree that lets us share states between different phones.
4. We use a dual decision tree that specifies 10K different Gaussian mixture models, but a total of 50K context-dependent states, each of which has its own mixture weights for one of the 10K GMMs.

The first three changes are tested on the DEV'07 set using speaker-adapted, vowelized models comprising 400K Gaussian mixture components trained on the GALE Phase 3.5 training set, which contains 1500 hours of audio. As can be seen in Table 2, we achieve small improvements with each change to the context modeling. The dual decision tree is compared to a standard decision tree on the same task, but with models comprising 800K mixtures that were trained on the full 1800 hours. As can be seen in Table 3, the dual tree's more detailed acoustic modeling further improves performance.

4. NEURAL NETWORK FEATURES

The neural network features used in IBM's 2009 evaluation system follow the Tandem [17] model. A multilayer perceptron (MLP) is trained to discriminate between 36 different phonetic targets. The input to the MLP is 9 successive frames (±4 frames around the current frame) of 13-dimensional VTLN PLP features plus delta and double-delta features, for a total of 351 input units. The PLP features were mean- and variance-normalized per speaker. There is a single hidden layer comprising 10,000 hidden units that use a logistic nonlinearity. The 36 output units are processed through a softmax nonlinearity, and the MLP is trained with the cross-entropy error criterion on 760 hours of GALE Arabic data.

To provide features for a standard GMM/HMM acoustic model, the MLP outputs are processed through a logarithmic nonlinearity, so the features are estimates of phone log-posterior probabilities, given the acoustic data. Unlike the standard Tandem approach, the features were not orthogonalized using the Karhunen-Loeve Transform (KLT). Instead, we train a global, semi-tied covariance (STC) transform [5], interleaving GMM and STC updates during model training. Because a global STC transform attempts to diagonalize the class-conditional feature distributions instead of the global feature distribution, we expected that this approach would outperform diagonalization based on the KLT. Pilot experiments on a 50-hour subset of the GALE Arabic training data confirmed this expectation.

Model	DEV'07	EVAL'08U	EVAL'09
V	8.2%	7.9%	9.7%
V-NN	8.1%	7.9%	9.6%

Table 4. % word error rates for the **V** and **V-NN** models on DEV'07, EVAL'08U, and EVAL'09, with Model M rescoring.

	DEV'07	EVAL'08U	EVAL'09
Baseline LM	8.4%	8.6%	10.0%
Model M	7.9%	8.0%	9.6%
Syntax	7.6%	7.6%	9.3%

Table 5. Word error rates after Model M and syntax rescoring for the SGMM system.

To illustrate the benefits of the neural network features, we compare the performance of the **V** model alone to the multi-stream **V-NN** model in Table 4. Both models are cross-adapted on the **SGMM** output and the resulting lattices are rescored with the Model M language model (Section 5), so the **V-NN** results match those for Step 13 in Table 1. We see small improvements on two of the three data sets. In designing the final system combination, we also observed that combinations using the **V-NN** model consistently outperformed combinations using only the **V** model, but that the difference was small.

5. MODEL M LANGUAGE MODELS

Model M is a novel, class-based exponential language model. It is motivated by the observation that shrinking the sum of parameter magnitudes in an exponential language model tends to improve performance [1]. As mentioned in Section 2.2, the baseline language model is a linear interpolation of 4-gram models built on 20 different sources. We build Model M models on the nine corpora with the highest interpolation weights in the baseline model, and create a new language model which is an interpolation of the nine Model M language models and the original unpruned 883M n-gram language model. We use this new language model to rescore the lattices for all the systems used in the evaluation. In the first two rows of Table 5, we present lattice rescoring results for the baseline language model and this new language model for the SGMM system. We see that significant improvements of 0.4-0.6% absolute are achieved. Similar improvements are obtained for all the other acoustic models.

6. NEURAL NETWORK SYNTACTIC LANGUAGE MODELS

We incorporate syntactic and morphological features as additional context features in a neural network language model (NNLM), and obtain up to 5% relative word error rate improvement over a regular word-based NNLM. Details are published in [2], while here we provide a brief summary and new results on multiple systems used in the 2009 GALE (P4) evaluation, in combination with Model M.

Long-span context words and syntactic features from the parse tree may be complementary to n-gram features, and are easily incorporated into a NNLM. The syntactic features we use include exposed head words and their non-terminal labels, both before and after the predicted word. Before parsing, we segment words into morphs using Arabic Treebank (ATB) segmentation. We use a maximum entropy parser trained on the Arabic Treebank and various broadcast news data sources released under the DARPA GALE program. After parsing, exposed head words and their non-terminal labels are extracted as additional context features used by the NNLM.

During testing, we use an N-best rescoring framework to take advantage of the complete parse tree of the entire sentence, as well as different tokenization representations (words versus morphs), by combining scores at the sentence level. The best weights of different models are found by simplex optimization on DEV'07. We choose $N = 100$ because the 100-best oracle word error rate is very close to that of the lattice.

The syntactic NNLM is trained on a subset of the language model training data deemed most relevant to the task: the 15 million words of audio transcripts. For the NNLM, we use 30 dimensional input vectors, 100 hidden units, and up to 60 epochs of training. The NNLM provides probabilities for only the 20k most frequent morphs, and we normalize using a background 4-gram language model trained on the ATB segmented audio transcripts.

Using syntactic NNLMs, we had previously obtained 0.3-0.5% absolute word error rate improvement over a regular word-based 6-gram NNLM on the best set of lattices from the 2008 (P3.5) evaluation. For the 2009 (P4) evaluation, we observe that combining a word-based NNLM with Model M gives only small improvements. For example, adding the NNLM to Model M results in word error rates of 7.8% for DEV'07 and 8.0% for EVAL'08. As shown on the last line in Table 5, NNLMs with morphological and syntactic features make further improvement when combined with Model M, up to 0.4% absolute improvement. Similar improvements are observed for other test sets and acoustic models. For example, with the unvowelized (U) system, the improvement on EVAL'09 is 0.3%.

7. SUMMARY

In this paper we have presented IBM's 2009 GALE Arabic speech transcription system, and described improvements made over the past year that led to a word error rate of 8.9% on the 2009 evaluation data and a year-to-year, absolute reduction of 1.6% word error rate on the unsequestered 2008 evaluation data. The key advances that enabled this level of performance are improvements to context-dependent modeling in vowelized Arabic acoustic models; the use of neural-network features provided by ICSI; Model M language models; a neural network language model that uses syntactic and morphological features; and improvements to our system combination strategy.

8. ACKNOWLEDGMENT

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

9. REFERENCES

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

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

[3] 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.

[4] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[5] 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.

[6] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech," in *Proc. ICASSP*, 1996, vol. I., pp. 339–341.

[7] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[8] 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.

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

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

[11] 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.

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

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

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

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

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

[17] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, vol. III, pp. 1635–1638.

[2] 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.