From Modern Standard Arabic to Levantine ASR: Leveraging GALE for Dialects

Hagen Soltau*, Lidia Mangu*, Fadi Biadsy†

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

***Abstract*—We report a series of experiments about how we can progress from Modern Standard Arabic (MSA) to Levantine ASR, in the context of the GALE DARPA program. While our GALE models achieved very low error rates, we still see error rates twice as high when decoding dialectal data. In this paper, we make use of a state-of-the-art Arabic dialect recognition system to automatically identify Levantine and MSA subsets in mixed speech of a variety of dialects including MSA. Training separate models on these subsets, we show a significant reduction in word error rate over using the entire data set to train one system for both dialects. During decoding, we use a tree array structure to mix Levantine and MSA models automatically using the posterior probabilities of the dialect classifier as soft weights. This technique allows us to mix these models without sacrificing performance for either varieties. Furthermore, using the initial acoustic-based dialect recognition system's output, we show that we can bootstrap a text-based dialect classifier and use it to identify relevant text data for building Levantine language models. Moreover, we compare different vowelization approaches when transitioning from MSA to Levantine models.**

I. Introduction

One of the key challenges in Arabic speech recognition research is how to handle the differences between Arabic dialects. Most recent work on Arabic ASR have addressed the problem of recognizing Modern Standard Arabic (MSA). Little work has focused on dialectal Arabic [1], [2]. Arabic dialects differ from MSA and each other in many dimensions of the linguistic spectrum, morphologically, lexically, syntactically, and phonologically. What makes Arabic dialect challenging in particular is the lack of a well-defined spelling system, resources (i.e., acoustic and LM training data) as well as tools (such as morphological analyzers and disambiguation tools).

In this paper, we build an Arabic ASR system that can handle Levantine dialects as well as MSA, by building Levantine/MSA-specific models. To do that, we first make use of an automatic dialect recognition system to annotate our GALE acoustic data with dialect IDs. We also build a text-based dialect ID system to identify Levantine LM training data in our pool of text data. For pronunciation modeling, we experiment with multiple techniques to compensate for the lack of a Levantine morphological analyzer.

The details of our GALE system are described in [3], [4], [5]. The acoustic training data consists of 1800 hours of broadcast news and conversations. Briefly, we use the following set of acoustic models, employing different pronunciation modeling approaches for Arabic:

- *Unvowelized:*
  This is a letter-to-sound mapping; short vowels and other diacritic markers are ignored. While these models don't perform as well as vowelized models at the ML level, discriminative training reduces the gap to a very large extent.
- *Buckwalter Vowelized:*
  The Buckwalter morphological analyzer [6] is used to generate candidates of vowelized (diacritized) pronunciations in a context independent way. This is also a letter-to-sound mapping, but we map each letter *and* diacritic marker (except for *shadda* marker, where consonants are doubled) to a phoneme. On average there are 3.3 pronunciations per words. For decoding, we use pronunciation probabilities that are obtained from the training data.
- *MADA Vowelized:*
  As an alternative to Buckwalter, we use MADA [7] to generate context-dependent diacritized candidates. In this approach, we apply the 15 linguistically-motivated pronunciation rules, to map diacritized words to true phonemes, described in [8]. On average there are 2.7 pronunciations per words. Pronunciation probabilities are derived from the MADA output. The details of generating the training and decoding pronunciation lexicons for this system are described in [9].

All models are based on context expanded PLP features with cepstral mean and variance normalization (CMVN) plus LDA and STC. Speaker adaptation consists of VTLN, FMLLR, and MLLR regression trees. Discriminative training uses both feature space (fBMMI) and model space (BMMI) training. A more detailed description of our training recipe and toolkit can be found in [10].

We use a 795K word vocabulary, which has OOV rates of less than 1% for all the GALE test sets. The language model is an unpruned 4-gram with 913M n-grams.

The paper is organized as follows. First, we describe the dialect recognition system, which enables us to define a Levantine test set (verified manually by the LDC). Second, we compare different pronunciation modeling strategies on the task of Levantine ASR. We then use the dialect classifier to find relevant subsets of our training data (both acoustic and text) to improve our models on Levantine.

II. Dialect Identification

As mentioned above, we are interested in building Levantine-specific models using the available GALE data. Recall that this data contains a mix of dialects in addition to MSA and that this data has no specific dialect annotations. To build a Levantine-specific ASR system, we need dialect annotations for each utterance since Arabic speakers, in broadcast conversations (BC), tend to code mix/switch between MSA and their native dialects across utterances and even within the same utterance.[1] In this work, we build a dialect recognition system to identify dialects at the utterance level.

Biadsy et al. [9], [11] have previously shown that a dialect recognition approach that relies on the hypothesis that certain phones are realized differently across dialects achieve state-of-the-art performance for multiple dialect and accent tasks (including Arabic). In this paper, we make use of this system (described next) to annotate some of our Arabic GALE data.

A. Phone Recognizer and Front-End

The dialect recognition approach makes use of phone hypotheses. Therefore, we first build a triphone CD- phone recognizer. The phone recognizer is trained on MSA using 50h of GALE speech data of broadcast news and conversations with a total of 20,000 Gaussians. We use one acoustic model for silence, one for non-vocal noise and another to model vocal noise. The front-end is a 13-dimensional PLP front-end with CMVN. Each frame is spliced together with four preceding and four succeeding frames and then LDA is performed to yield 40d feature vectors. We utilize a unigram phone model trained on MSA to avoid bias for any particular dialect.[2] We also use FMLLR adaptation using the top CD-phone sequence hypothesis. Our phone inventory includes 34 phones, 6 vowels and 28 consonants.

B. Phone GMM-UBM and Phonetic Representation

The first step in the dialect recognition approach is to build a 'universal'acoustic model for each context-independent phone type. In particular, we first extract acoustic features (40d feature vectors after CMVN and FMLLR) aligned to each phone instance in the training data (a mix of dialects). Afterwards, using the frames aligned to the same phone type (in all training utterances), we train a Gaussian Mixture Model (GMM), with 100 Gaussian components with diagonal covariance matrices, for this phone type, employing the EM algorithm. Therefore, we build 34 GMMs. Each phone GMM can be viewed as a GMM-Universal Background Model (GMM-UBM) for that phone type, since it models the general realization of that phone across dialect classes [12]. We call these GMMs *phone GMM-UBMs*.

Each phone type in a given utterance (U) is represented with a single MAP (Maximum A-Posteriori) adapted GMM. Specifically, we first obtain the acoustic frames aligned to every phone instance of the same phone type in U. Then these frames are used to MAP adapt the means of the corresponding phone GMM-UBM using a relevance factor of $r = 0.1$. The resulting GMM of phone type ϕ is called the *adapted phone-GMM* (f_ϕ). The intuition here is that f_ϕ 'summarizes' the variable number of acoustic frames of all the phone instances of a phone-type ϕ in a new distribution specific to ϕ in U [11].

C. A Phone-Type-Based SVM Kernel

Now, each utterance U can be represented as a set S_U of adapted phone-GMMs, each of which corresponds to one phone type. Therefore, the size of S_U is at most the size of the phone inventory ($|\Phi|$). Let $S_{U_a} = \{f_\phi\}_{\phi\in\Phi}$ and $S_{U_b} = \{g_\phi\}_{\phi\in\Phi}$ be the adapted phone-GMM sets of utterances U_a and U_b, respectively. Using the kernel function in equation (1), designed by [11] which employed the upper bound of KL-divergence-based kernel (2), proposed by [13], we train a binary SVM classifier for each pair of dialects. This kernel function compares the 'general' realization of the same phone types across a pair of utterances.

$$K(S_{U_a}, S_{U_b}) = \sum_{\phi\in\Phi} K_\phi(f'_\phi, g'_\phi) \tag{1}$$

where f'_ϕ is the same as f_ϕ but we subtract from its Gaussian mean vectors the corresponding Gaussian mean vectors of the phone GMM-UBM (of phone type ϕ). g'_ϕ is obtained similarly from g_ϕ. The subtraction forces zero contributions from Gaussians that are not affected by the MAP adaptation. And,

$$K_\phi(f_\phi, g_\phi) = \sum_i (\sqrt{\omega_{\phi,i}}\Sigma_{\phi,i}^{-\frac{1}{2}}\mu_i^f)^T(\sqrt{\omega_{\phi,i}}\Sigma_{\phi,i}^{-\frac{1}{2}}\mu_i^g) \tag{2}$$

where, $\omega_{\phi,i}$ and $\Sigma_{\phi,i}$ respectively are the weight and diagonal covariance matrix of Gaussian i of the phone GMM-UBM of phone-type ϕ; μ_i^f and μ_i^g are the mean vectors of Gaussian i of the *adapted* phone-GMMs f_ϕ and g_ϕ, respectively.

It is interesting to note that, for (1), when K_ϕ is a linear kernel, such as the one in (2), each utterance S_{U_x} can be represented as a single vector. This vector, say W_x, is formed by stacking the mean vectors of the adapted phone-GMM (after scaling by $\sqrt{\omega_\phi}\Sigma_\phi^{-\frac{1}{2}}$ and subtracting the corresponding μ_ϕ) in some (arbitrary) fixed order, and zero mean vectors for phone types not in U_x. This representation allows the kernel in (1) to be written as in (3). This vector representation can be viewed as the 'phonetic finger print' of the speaker. It should be noted that, in this vector, the phones constrain which Gaussians can be affected by the MAP adaptation (allowing comparison under linguistic constraints realized by the phone recognizer), whereas in the GMM-supervector approach [14], in theory, any Gaussian can be affected by any frame of any phone.

$$K(S_{U_a}, S_{U_b}) = W_a^T W_b \tag{3}$$

[1] In this work, we do not attempt to identify code switching points; we simply assume that an utterance is spoken either in MSA or in purely a regional dialect.

[2] We use true phonetic labels here by generating pronunciation dictioanries using MADA, following [9].

III. ASR and Dielct ID Data Selection

As noted above, the GALE data is not annotated based on dialects. Moreover, to the best of our knowledge, there is no Arabic dialect corpus of similar domain and/or acoustic condition as BC. Fortunately, there are telephone conversation corpora available from the LDC for four Arabic dialects (Egyptian, Levantine, Gulf, and Iraqi). To address the acoustic recording and domain issues we build two systems.

In our first system, we train our dialect recognition on dialect data taken from spontaneous telephone conversations from the following Appen corpora: Iraqi Arabic (478 speakers), Gulf (976), and Levantine (985). For Egyptian, we use the 280 speakers in CallHome Egyptian and its supplement. We train the system on 30 s cuts. Each cut consists of consecutive speech segments totaling 30 s in length (after removing silence). Multiple cuts are extracted from each speaker. [3]

We run this system to annotate a portion of our GALE BC data (after downsampling to 8Khz). The dialect recognition system classified 54h of Levantine speech with a relatively high confidence. Since the dialect ID system is trained on telephone conversations as opposed to broadcast conversations, we asked the LDC to validate/filter the output of the system. We find that about 36h out of 54h are tagged as "mostly Levantine", a 10h set contains code switching between MSA and Levantine at the utterance level, and an 8h set contains either other dialects or MSA. Recall our system is not trained to identify MSA.

We extract a 4h test set (**LEV_4h**) to be used for reporting results in all the Levantine ASR experiments. From the remaining 32h we extract all the utterances longer than 20 seconds, this yields approximately 10h of data (**LEV_10**). Part of the transcripts released by LDC for the GALE program have "non-MSA" annotations. This allows us to select a 40h MSA corpus by choosing speakers whose utterances have no such markings. From this set we select 4h for our MSA ASR experiments (**MSA_4h**). From the remining, we further select a 10h set with utterances longer than 20 seconds (**MSA_10**).

IV. Dialect Identification on GALE data

Given that now we have gold standard BC MSA and Levantine data (MSA_10 and LEV_10), we can train another dialect recognition system to distinguish MSA vs. Levantine for BC acoustic conditions. We divide LEV_10 into 9h for training and 1h for testing our dialect recognition system. Similarly MSA_10 is divided into 9h for training and 1h for testing. Note that this amount of acoustic data is typically not sufficient to train dialect identification systems; however, we are interested in making use of the rest of the data for other experiments.

As described in Section II, for the dialect identification system we need a phone decoder; therefore we carry out a number of experiments for finding the best strategy for building it. We train 3 MADA Vowelized (i.e., a true phonetic-based system) triphone acoustic models in which we vary the number of Gaussians and the number of states, using either ML or discriminative training. First, we test these models for word recognition with our unpruned 4-gram LM. Table I shows the word error rates on the DEV-07 set.

System	WER on DEV-07
50k Gaussians, 1k states, ML	16.8%
200k Gaussians, 5k states, ML	15.4%
200k Gaussians, 5k states, fBMMI+BMMI	12.5%

TABLE I
MADA AM used for Dialect ID, WER test

System / Features	Classificaion Accuracy
50k ML 1-gram phone LM	85.1%
50k ML 3-gram phone LM	84.5%
200k ML, 3-gram phone LM	84.9%
200k fBMMI+BMMI, 3-gram	83.0%

TABLE II
Dialect classification performance

In the next test, we use the triphone models to decode phone sequences with different phone language models. For each phone decoder we build a dialect classification system using the SVM-Kernel approach described in Section II-C. We train the models on 9h of Levantine data and 9h of MSA data, and evaluate the results on a test set which contains 1h of Levantine and 1h of MSA data. Table II shows the dialect classification rates for the different acoustic model and phone language model combinations. Based on these results we decided to use the smallest, simplest model (50K Gaussians ML model with unigram phone language model) for the subsequent experiments.

V. Acoustic Modeling Experiments

A. Comparing Vowelizations

We select a 300 hour subset from our entire GALE training set and train speaker adaptive acoustic models for all 3 lexical setups. The decoding setup includes VTLN, FMLLR, and MLLR and we use an unpruned 4-gram LM with a 795k vocabulary. First, we test the models on one of our standard GALE development sets, DEV-07, shown in table III. Buckwalter and MADA vowelizations perform similarly, while the unvowelized models are 2.7% worse at the ML level. However, we want to note that the difference is only 1% after discriminative training. This indicates that discriminative training of context-dependent (CD) GMM models is able to compensate for the lack of (knowledge based) pronunciation modeling to a large degree.

In the next comparison, we test the models on a newly defined MSA test set. The reason behind this set is that we want to use the same methodology for defining/selecting a test set for both Levantine and MSA. We would like to analyze the difficulty of Levantine when compared to MSA under exactly

[3]The equal error rate reported by Biadsy et al. [11] of this dialect recognition system on a 20% held-out speaker set from these corpora is 4%.

System	Unvowelized	BW Vowelized	MADA Vowelized
ML	16.6%	14.2%	13.9%
fBMMI+BMMI	12.7%	11.8%	11.7%

TABLE III
300H AM TESTED ON DEV-07

System	Unvowelized	BW Vowelized	MADA Vowelized
ML	28.6%	27.0%	25.7%
fBMMI+BMMI	21.8%	21.7%	21.2%

TABLE IV
300H AM TESTED ON MSA_4H

System	Unvowelized	BW Vowelized	MADA Vowelized
ML	48.2%	50.3%	48.1%
fBMMI+BMMI	39.7%	42.1%	40.8%

TABLE V
300H AM TESTED ON LEV_4H

Training data	WER
unweighted (300h)	48.2%
hard-weighted (37h)	48.3%
soft-weighted (300h)	45.3%

TABLE VI
COMPARING WEIGHTING SCHEMES OF TRAINING STATISTICS ON LEV_4H, 300H SETUP, UNVOWELIZED ML MODELS

same conditions. We are basically reducing effects related to how and from where the test sets are chosen. DEV-07, for example, is a test set defined by LDC which consists of mostly very clean broadcast news data. This is very likely the reason behind our very low error rates. The MSA_4h test set is selected randomly from broadcast conversations of our training set and labeled as MSA by our dialect classifier. The reason to select the data from broadcast conversations is to match the conditions of the Levantine test set. All of the Levantine data comes from BC as well. The error rates on this MSA test set (Table IV) is almost twice as high as the error rates on DEV-07 (Table III), although both are non-dialectal (MSA) test data. We also see that all three models perform at a similar level (21.2% - 21.8%) after discriminative training.

We now compare the models on Levantine data (LEV_4). Recall that this Levantine test set is part of the GALE corpus identified automatically by our dialect classifier and manually verified by LDC (see Section II). The same methodology for selecting the test data is used for MSA_4h and LEV_4h. Both MSA_4h and LEV_4h test sets are excluded from the training of the acoustic and language models. Looking at Tables IV and V, we observe two main points:

1) The error rate for Levantine is almost twice as high as for MSA (39.7% vs 21.8%). We compare here the Levantine error rate to MSA_4h and not to DEV-07. This allows us to attribute the increase in error rate to dialect and not to other effects (how the test set was chosen and how carefully the transcripts were done).
2) Another interesting observation is that the unvowelized models perform best on Levantine (39.4% vs. 40.8% and 42.1%). We speculate that this due to the fact that both Buckwalter analyzer, MADA, and the pronunciation rules are designed for MSA – which do not work properly for Levantine words. A dialect specific morphological analyzer would very likely improve results, but it is unclear that it would significantly reduce the error rate on Levantine given that the unvowelized perform comparably well on MSA data (Table IV).

B. Selecting dialect data from the 300 hour training subset

We now run the dialect recognition system on our 300 hours, a subset of the GALE training corpus. Out of this training set, we obtain about 37 hours labeled as Levantine. This is not sufficient to train a set of acoustic models. One option is to use a deep MLLR regression tree or MAP training. In our experience MLLR works well for limited domain adaptation data, but will not be able to fully utilize a large amount of domain adaptation data. While MAP works better with more adaptation data, it is difficult to use it in combination with feature space discriminative training.

Instead, we use a form of training with weighted statistics. The advantage is that all components of the model (including decision trees) are trained at all training stages (ML, DT) with the new domain data. In our case we have additional information in form of a dialect posterior probabilities for each utterance from the dialect classifier. We use this posterior to weight the statistics of each utterance during ML and discriminative training.

Table VI shows a comparison of different weighting schemes. In the first row, we simply train on all 300 hours regardless whether they are Levantine or MSA. This model gives us an error rate of 48.2%. In the second row, we train only on the selected Levantine subset of 37 hours. The error rate is slightly higher, 48.3%, due to the lack of training data. In the third row, we train on the same 300 hours, but weight the statistics of each utterance individually by the posterior score of the dialect classifier. This provides us with a smoothing of the models, avoids overtraining and we get a 2.9% error reduction.

We apply now the soft-weighting scheme to all vowelization setups and compare the models both after ML and fBMMI+BMMI training in Table VII. The improvement from focusing on Levantine training data can be seen by comparing Table V with Table VII. For example, for the unvowelized models, we obtain 2.9% absolute error reduction at the ML level, and 1.3% after discriminative training. Note that we do not add training data, rather we find relevant subsets that match our target dialect.

C. Tree array combination

When we focus the training on Levantine, we can expect the model to perform worse on MSA data. In fact, the error

System	Unvowelized	BW Vowelized	MADA Vowelized
ML	45.3%	47.3%	45.5%
fBMMI+BMMI	38.4%	41.4%	39.2%

TABLE VII
300H AM TESTED ON LEV_4H

Weight for MSA model	Weight for LEV models	DEV-07	LEV_4h
1.0	0.0	12.7%	39.7%
0.0	1.0	15.1%	38.4%
0.5	0.5	13.3%	38.2%
dialect Classifier soft weight		12.9%	38.4%

TABLE VIII
TREEARRAY COMBINATION OF GENERAL MODELS WITH LEVANTINE MODELS IN ON DECODING PASS, 300H UNVOWELIZED FBMMI+BMMI SETUP

Training data	WER
unweighted (1800h)	47.0%
hard-weighted (237h)	42.3%
soft-weighted (1800h)	43.5%

TABLE IX
COMPARING WEIGHTING SCHEMES OF TRAINING STATISTICS ON LEV_4H, 1800H SETUP, UNVOWELIZED ML MODELS

Test data	dialect classification
MSA_4h	86.0%
Lev_4h	87.2%

TABLE X
TEXT ONLY DIALECT CLASSIFICATION USING LEVANTINE AND MSA LANGUAGE MODELS

rate increases from 12.7% to 15.1% on DEV-07 when we use the Levantine models (Table VIII). Our toolkit allows us to combine models with different decision trees into one single decoding graph [3]. This enables us to combine different acoustic models in one decoding pass on the fly, without making a hard model selection. The combined acoustic score is the weighted sum of the log likelihoods of the combined models. In our case, we combine the MSA and LEV unvowelized models. The results are in Table VIII. The first two rows represent the extreme cases where either the MSA or LEV model is used exclusively. In the third row, we weight both models equally and constant for all utterances. The error rate on DEV-07 is 13.3%, 0.6% higher than when just using the MSA model, but much better than when using the LEV models only (15.1%). On the other hand, we get a small improvement on the Levantine test set (38.4% goes to 38.2%). This is a system combination effect. We used tree arrays in the past as an alternative to rover or cross-adaptation, for example in our latest GALE evaluation. In the last row in Table VIII we use the posterior of the dialect classifier as a soft weight for model combination on a per utterance basis. This automatic strategy gives us an error rate that is close to the optimal performance of a model selected manually.

D. Selecting dialect data from the 1800 hour training set

The full GALE training corpus consists of about 1800 hours. Similar to the previous experiments, but now focusing exclusively on the unvowelized models, we generate dialect labels for the entire training corpus. The dialect recognition system identified about 237 hours as Levantine in the GALE corpus (or 13%). In Table IX, we compare different weighting schemes for the Levantine data. In contrast to the 300 hours setup (Table VI), the best error rate is achieved now by training exclusively on the 237 hours Levantine data and not by using the dialect scores to weight the statistics. The reason is simply that the amount of Levantine training data is now large enough to train acoustic models and we do not need to add data as it was the case for the previous experiments when we had only 37 hours of Levantine data.

After discriminative training (fBMMI+bMMI) of the 237 hours unvowelized Levantine models, the error rate goes down to 36.3%. In other words, we can lower the error rate by almost 10% relative by focusing on relevant subsets of the training data and the dialect classifier together with the tree array decoding technique which allows us to use both Levantine and MSA models in one decoding pass, so the engine can handle both dialectal and non-dialectal utterances at the same time.

VI. LANGUAGE MODELING EXPERIMENTS

A. Dialect ID based on Text only

The previous experiments in Section V demonstrate that the acoustic training data contains relevant dialect subsets when detected can improve the acoustic models. In this section, we report on a similar strategy for language modeling, but now we built a dialect classifier based on text only – no audio data is used. First, we build a Kneser-Ney smoothed 3-gram Levantine LM on the 2M words corresponding to the transcripts of the 237 hours Levantine acoustic training data (identified automatically). Similarly, we build an MSA language model from all the utterances which are classified as MSA with more than 95% probability by the dialect annotator. We build a text dialect classifier which simply checks the log-likelihood ratio of the two LMs on a given utterance. Table X shows that we can predict the dialect reliably even when only text data is available.

B. Levantine LM

Our GALE language models are trained on a collection of 1.6 billion words, which we divide into 20 parts based on the source. We train a 4-gram model with modified Kneser-Ney smoothing [15] for each source, and then linearly interpolate the 20 component models with the interpolation weights chosen to optimize perplexity on a held-out set. In order to build a Levantine language model, we run the text dialect annotator described above on each of the 20 text sources and build 4-gram language models on the 20 dialectal subparts. The new 20 dialect language models are interpolated with the 20 original ones. We optimize the interpolation weights of

Training data	WER
913m 4-gram baseline LM	36.3%
+ 3-gram Levantine LM from 238h set	35.4%
+ 4-gram Levantine weighted LM (all text sources)	35.1%

TABLE XI
LM RESCORING WITH LEVANTINE LM

the 40 language models on a levantine held-out set. Table XI shows the improvements obtained by adding dialect data to the original language model. Note that the improvement from adding dialect language models is less than the one obtained from dialect acoustic models. One reason for this is the fact that the initial dialect data is selected from the BC part of the training data, and the BC language model has a high weight in the baseline interpolated LM.

C. Finding Levantine words

We can identify dialectal words if we compute how many times the word occurs in the Levantine corpus vs. the MSA one. After sorting the count ratios, we find the following words ranked at the top of the list: Em, hyk, bdw, bdk, ylly, blbnAn, which are in fact Levantine words. Note that identifying dialectal words can be useful for building better pronunciation dictionaries for dialects as well as for machine translation.

VII. CONCLUSION

In summary, this paper has presented a series of experiments to leverage our GALE acoustic and language models for Levantine. The dialect recognition system allows us to focus on relevant training subsets and improves the models by almost 10% relative. While specialized models for Levantine perform poorly on MSA, the tree array decoding procedure allows us to mix both models without sacrificing performance. We compared different vowelization strategies and showed that the unvowelized models are very competitive after discriminative training. Also, we showed that we can build a text-only dialect classifier that performs as well as a dialect classifier requiring audio data. The text only dialect classifier enables us to find relevant LM text data. In the future we plan to apply this classifier on parallel text corpora for machine translation to leverage GALE MT for Levantine. Future work will also include pronunciation modeling for Levantine where the text based dialect classifier can provide us with candidate words that occur only in Levantine. Our text-based dialect classifier can also be improved by employing discriminative classifiers (such as, logistic regression and SVM) instead of likelihood ratios.

ACKNOWLEDGMENT

We would like to acknowledge the support of DARPA under Grant HR0011-06-2-0001 for funding part of this work. We thank the LDC for annotating the dialect test corpus. We thank Jason Pelecanos for useful discussions regarding UBM models for dialect recognition.

REFERENCES

[1] K. Kirchhoff and D. Vergyri, "Cross-Dialectal Acoustic Data Sharing for Arabic Speech Recognition," in *ICASSP*, 2004.
[2] D. Vergyri, K. Kirchhoff, R. Gadde, A. Stolcke, and J. Zheng, "Development of a Conversational Telephone Speech Recognizer for Levantine Arabic," in *Interspeech*, 2005.
[3] H. Soltau, G. Saon, B. Kingsbury, H.-K. J. Kuo, L. Mangu, D. Povey, and A. Emami, "Advances in Arabic speech transcription at IBM under the DARPA GALE program," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 17, no. 5, pp. 884–894, 2009.
[4] B. Kingsbury, H. Soltau, G. Saon, S. Chu, H.-K. J. Kuo, L. Mangu, S. Ravuri, N. Morgan, and A. Janin, "The IBM 2009 GALE Arabic speech transcription system," *Proc. ICASSP*, 2011.
[5] L. Mangu et. al., "The IBM 2011 GALE Arabic speech transcription system," in *Proc. of IEEE ASRU*, 2011, Submitted.
[6] T. Buckwalter, "LDC2004L02: Buckwalter Arabic morphological analyzer version 2.0," 2004.
[7] Nizar Habash and Owen Rambow, "Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, June 2005, pp. 573–580, Association for Computational Linguistics.
[8] 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.
[9] Fadi Biadsy, "Automatic Dialect and Accent Recognition and its Application to Speech Recognition," in *PhD. Thesis, Columbia University*, 2011.
[10] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEE Workshop on Spoken Language Technology*, 2010.
[11] F. Biadsy, J. Hirschberg, and D. Ellis, "Dialect and Accent Recognition using Phonetic-Segmentation Supervectors," in *Interspeech*, 2011.
[12] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, 2000.
[13] W.M. Campbell, D.E. Sturim, D.A. Reynolds, and A. Solomonoff, "SVM Based Speaker Verification Using a GMM Supervector Kernel and NAP variability compensation," in *Proceedings of ICASSP'06*, France, May 2006.
[14] W.M. Campbell, D.E. Sturim, and D.A. Reynolds, "Support Vector Machines Using GMM Supervectors for Speaker Verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, May 2006.
[15] S. F. Chen and J. T. Goodman, "An empirical study of smoothing techniques for language modeling," Tech. Rep. TR-10-98, Harvard University, 1998.