GOOGLE'S CROSS-DIALECT ARABIC VOICE SEARCH

Fadi Biadsy, Pedro J. Moreno, Martin Jansche

Google Inc.
New York, NY, USA
{*biadsy, pedro, mjansche*}*@google.com*

ABSTRACT

We present a large scale effort to build a commercial Automatic Speech Recognition (ASR) product for Arabic. Our goal is to support voice search, dictation, and voice control for the general Arabic-speaking public, including support for multiple Arabic dialects. We describe our ASR system design and compare recognizers for five Arabic dialects, with the potential to reach more than 125 million people in Egypt, Jordan, Lebanon, Saudi Arabia, and the United Arab Emirates (UAE). We compare systems built on diacritized vs. non-diacritized text. We also conduct cross-dialect experiments, where we train on one dialect and test on the others. Our average word error rate (WER) is 24.8% for voice search.

Index Terms— Speech Recognition, Arabic, Voice Search

1. INTRODUCTION

The Arabic language has multiple variants, including Modern Standard Arabic (MSA), the formal written standard language of the media, culture and education across the Arab world. MSA however is not a native language of any Arab. The Arabic dialects, in contrast, are the true native language forms. They are generally restricted in use to informal daily communication. They are not taught in schools or even standardized, although there is a rich popular dialect culture of folktales, songs, movies, and TV shows. Dialects are primarily spoken, not written. However, this is changing as more Arabs gain access to electronic media such as social networks, email, etc.

The Arabic dialects we see today originate from historical interactions between Classical Arabic and languages of the contemporaneous cultures. For example, Algerian Arabic has many influences from Berber as well as French. Arabic dialects differ substantially from MSA and each other in terms of phonology, morphology, lexical choice, and syntax. Arabic dialects are natively spoken by over 200 million people.

There has been a great deal of work on Arabic ASR; a considerable portion of this work has been done throughout the DARPA GALE project. However, the vast majority of these works have focused only on MSA, mostly for Broadcast News (BN) and Conversation (BC) (e.g., [1]). Dialectal Arabic has received less attention. Recent work on dialects includes a demonstration that some acoustic data from MSA slightly improves the recognition of Egyptian Arabic [2]; Biadsy [3] and Soltau et al. [4] have shown how an Arabic dialect ID system can be used to identify Levantine Arabic data in a mix of dialects in the GALE BC data to bootstrap a Levantine recognizer.

Here we describe Google's search by voice system for Arabic. In this system, users speak their search queries, typically using a mobile phone, and the system returns a transcription and web search results. We focus on five Arabic dialects collected from five countries: (1) Saudi-Arabia (collected from Riyadh and Jeddah) (SA), (2) the United Arab Emirates (Abu Dhabi and Dubai) (AE), (3) Jordan (Amman) (JO), (4) Lebanon (Beirut) (LB), and (5) Egypt (Cairo) (EG). The paper is organized as follows. We describe our data collection in Section 2. We then describe our system design in Section 3, where we also discuss our diacritized vs. non-diacritized systems. In Section 4 we show our experimental results on the five dialects as well as our cross-dialect comparisons. We conclude in Section 5.

2. DATA COLLECTION

Most of the available Arabic acoustic data employed by the ASR community is MSA recorded from BN and BC. As mentioned above, MSA is not the native language of Arabic speakers. Also, BN is typically well-planned read speech recorded in an environment free from background noise. The available dialectal Arabic data on the other hand are telephone conversations sampled at 8kHz. These genres and acoustic conditions do not meet our requirements for building an Arabic voice search system, for the following reasons: (1) Our data has to be recorded from different dialects across the Arab world to capture phonetic differences. (2) The data has to be classified based on these dialects so we can build and compare systems across dialects. (3) We prefer acoustic data sampled with 16kHz recorded using a mobile phone to avoid channel mismatch. (4) Our data should be recorded in a recording environment similar to that in which our system will be used (e.g., rooms, streets, shopping centers, cars, etc.). (5) The semantic nature of the utterance is also specific to the task. A voice search system expects short spoken search queries, not long dictation or BN-style utterances.

For all these reasons we decided that we needed to collect our own acoustic data from different parts of the Arab Wold. The efficient collection of high quality data thus became a crucial issue in our system development. We collected our spoken utterances in different acoustic environments from a variety of speakers using our DataHound Android application [5]. This application displays prompts based on common Arabic and English search queries on a mobile device. We asked our users to speak these queries as close to their native dialect as possible. We recruited native Arabic speakers from each of the five dialects to record more than a quarter million spoken queries per dialect. Our speakers are both female and male volunteers from different age ranges. We recorded the audio samples in both quiet and noisy environments, including offices, shopping centers, public transportation, and others. We use most of the data as training and hold out around 15,000 utterances for testing from each dialect. The sets of speakers appearing in the training and test sets are disjoint in all dialects. The number of speakers and queries and the total duration of our data sets are shown in Table 1.

Dialect	Set	# Speakers	# Queries	Duration
EG	train	604	245K	223 hrs
	test	29	15K	12.4 hrs
JO	train	848	260K	224 hrs
	test	21	15K	10.7 hrs
SA	train	745	299K	226 hrs
	test	29	15K	10.6 hrs
AE	train	587	235K	193 hrs
	test	29	15K	10.6 hrs
LB	train	795	264K	219 hrs
	test	29	15K	13.8 hrs

Table 1. Train and Test Data for All Dialects

3. ASR SYSTEM DESCRIPTION

3.1. Acoustic Modeling

The acoustic models in our systems are standard 3-HMM-state context-dependent (tri-phone) models with a variable number of Gaussians per state. The front-end is a 13-dimensional PLP front-end with cepstral mean normalization and energy-based endpointer, to remove excessive silence. Each frame is spliced together with four preceding and four succeeding frames and then Linear Discriminant Analysis (LDA) is performed to yield 39-dimensional feature vectors.

Our acoustic models are gender-independent, maximum-likelihood trained interleaved with estimation of a global semi-tied covariance (STC) transform, followed by boosted MMI (BMMI) training [6]. For decoding, we use a statically constructed context-dependent lexicon and language model (see below) which are expressed as finite-state transducers and composed on-the-fly [7]. We use CMLLR for speaker adaptation. The CMLLR matrix is estimated and incrementally refined using previous sessions (query utterances) of the same speaker.[1]

3.2. Language Modeling

In voice search, users speak their search queries and the system returns search results. We have observed that speakers formulate their spoken queries similarly to the way they type them. This observation is consistent across languages. Therefore, the best available source of training data for our Language Models (LM) are the typed search queries. Since textual data may vary across dialects, we train separate LMs on anonymized search query logs from the corresponding regional search engine. For example, the Egyptian Arabic LM is trained on the textual queries directed to *www.google.com.eg*. The LM for each dialect is trained on search queries sampled only from one year's worth of search query logs. In this work, we make use of 5-gram backoff LMs, trained with Katz smoothing [9] and entropy pruning [10]. All the systems described in this paper make use of a vocabulary size of one million lexical items.

3.3. Lexicon

For languages with complex letter-to-sound mappings, pronunciation dictionaries are typically written by hand. However, for morphologically rich languages, such as Arabic, pronunciation dictionaries are difficult to create by hand, because of the large number of word forms, each of which has a number of possible pronunciations. Arabic morphological features are realized using both concatenative (affixes and stems) and templatic (root and pattern) morphology with a variety of morphological and phonological adjustments that appear in word orthography and interact with orthographic variations.

Fortunately, the relationship between orthography and pronunciation is relatively regular and well understood for MSA. However, to be able to map Arabic words to their true phonemic representations, the words have to be first fully *diacritized* in order to resolve the ambiguities present in the conventional orthography. Diacritics in Arabic are used to unambiguously denote the presence or absence of short vowels, distinguish long vowels from glides or diphthongs, and indicate geminate consonants. Unfortunately, these diacritics are largely restricted to religious texts and MSA school textbooks. The system of MSA diacritics is not generally applicable to spoken dialects, which typically have richer vowel inventories. Furthermore, almost all our Arabic queries and training transcripts are completely undiacritized. In this work, we experiment with two approaches: a completely undiacritized ASR system and a fully diacritized one, employing a diacritization system.

3.3.1. Diacritized ASR System

For the diacritized ASR system, we first diacritize all our textual data (transcripts and LM data) using Google's Arabic diacritizer [11]. The diacritizer takes a sentence as input and determines the most likely diacritics for each word in context, based on word n-grams ($n = 1, 2, 3$) and letter n-grams ($n = 1, \ldots, 5$). Our diacritizer is trained on partially diacritized phrases or sentences crawled from the Web; therefore it is not tuned for any particular spoken Arabic dialect. Given the nature of diacritized Arabic text found on the Web, the diacritizer is biased towards Classical Arabic.

For this ASR system, we build our lexicon similar to [3, 12]. Specifically, employing the automatically diacritized textual data, we build our lexicon by mapping each fully diacritized word in our transcripts to its pronunciation using the pronunciation rules described in [3, 12]. Note that these rules are tuned for MSA, not for Arabic dialects. Our LM is also trained on fully diacritized texts.

3.3.2. Undiacritized ASR System

Both the lexicon and LM for the undiacritized ASR system are built using completely undiacritized texts.[2] The lexicon employs a simplistic letter-to-sound mapping (i.e., graphemic representation). Note that in this representation, short-vowels, for example, will not be modeled as separate phonemes; acoustically, they will be modeled as part of the surrounding consonant acoustic models. This treatment is similar to the "unvowelized" system in [4].

Two letters are typically confused in informal writing: *tā' marbūṭah* and final *hā'*. Both of these letters can be pronounced either /t/ or /h/. Note that the letter *hā'* in other positions in the word is always pronounced /h/. *'Alif maqṣūrah* (/a:/) is sometimes written instead of final *yā'* (/i:/). Instead of correcting the transcripts and adding pronunciation variants in the lexicon, we map these letters to two special phonemes. We hypothesize that the corresponding acoustic mixture models will capture their different realizations. The only phonological process we model in this lexicon is the 'sun-letters' assimilation (see [12]).

[1] This system is similar to our Cantonese voice search system [8].

[2] We removed any diacritics that may have been present.

System	WER (%)
Undiacritized	24.6
Fully Diacritized (score with diacritized)	29.3
Fully Diacritized (score with undiacritized)	27.5

Table 2. Diacritized vs. Undiacritized System for EG

System	Dialect-Specific	Combined
AE	27.7	29.7
SA	28.7	30.0
EG	24.6	29.8
JO	18.5	19.2
LB	24.2	27.9

Table 3. WER (%) of the Dialect-Specific vs. Combined Systems

3.3.3. English Words, Numbers, and URLs

Arabic typed search queries include a significant portion of tokens that are not written with Arabic script, including English words, numbers written with European digits, and URLs. Because such tokens are frequent, we include them in our LM and lexicon. The pronunciations of English words in our LM are imported from our American English lexicon. We map the English phonemes to the closest Arabic phonemes. URLs are initially identified in our LM data, and then parsed to tokens. Similar to English words, the pronunciations for these tokens are imported from the English lexicon. Each number in our LM data is first converted to a list of words using an Arabic number grammar (FST), and those words are then pronounced according to our Arabic pronunciation rules.

4. EXPERIMENTS

4.1. Diacritization or Not

We would like to determine whether diacritization is important for Arabic-dialect ASR. In this work, we test this hypothesis only for Egyptian Arabic. We train two Egyptian Arabic systems, as described in Section 3, using the Egyptian training data in Section 2. The first system employs fully diacritized components (lexicon and LM), as described in Section 3.3.1; and the second employs an undiacritized lexicon and LM, as explained in Section 3.3.2.

Note that the output transcripts of the diacritized system is fully diacritized text. To be able to score our utterances, we can initially either automatically diacritize our reference transcripts or remove diacritics from the recognition hypotheses.[3] In this work, we test both scoring schemes. Nevertheless, it is important to note that diacritics are typically not orthographically represented in Arabic texts. Diacritization is generally not necessary to make the transcript readable by Arabic literate readers. Thus, Arabic ASR systems typically do not output fully diacritized transcripts. Moreover, the vast majority of Arabic web pages are undiacritized, hence not useful for voice search. But, for other applications, such as speech-to-speech translation, a system that outputs fully diacritized text may benefit Arabic machine translation [13].

Table 2 compares the Word Error Rate (WER) of our undiacritized and diacritized Egyptian ASR system with both scoring schemes (tested on Egyptian Arabic). Interestingly, we observe that the undiacritized system outperforms the diacritized system with both scoring schemes. One possible explanation is that the automatic diacritizer we use here has not specifically tuned for diacritizing Egyptian Arabic, but rather a mix of Arabic Web text. Moreover, the pronunciation rules we employ to map from diacritized words to pronunciations have been designed for MSA. Our results are still consistent with previous work on Levantine Arabic [4], although our system is different in multiple aspects (e.g., the use of a fully diacritized LM). We also observe that, as expected, removing the diacritics improve WER, but it is interesting to see that the difference is not very large. In other words, our system can produce relatively accurate fully diacritized Arabic text.

[3]We do not apply any other Arabic text normalization steps before scoring, such as removing alef with hamza, etc.

4.2. Dialect-Specific Systems vs. One Combined System

After settling on an undiacritized system design based on the results of the comparison for Egyptian, we built a *dialect-specific* ASR system for each of the five dialects. Each dialect-specific system is trained on the corresponding training data, described in Section 2. Similarly, we also built a *combined* system trained on all the pooled acoustic data from the five dialects, and all the combined textual data for training the LM. All these systems employ the undiacritized lexicon and LM, as discussed in Section 3.3.2. The first column of Table 3 shows the WER of each dialect-specific system when evaluated on its held-out test data. The second column shows the WER of our combined system when tested on each dialect test set.

We observe that the best performing system is that of the Jordanian dialect followed by Lebanese (both are Levantine dialects), then Egyptian, followed by UAE Arabic, and Saudi Arabic dialect (UAE and Saudi Arabic are Gulf dialects). We have investigated the reason behind the relatively low WER for Jordanian Arabic. Listening to a sample of utterances of this dialect, we have noticed that the training and testing data sets were generally recorded in significantly less noisy environment than those of the other dialects. Further research is required to understand why the speech of Gulf dialects appears to be significantly harder to recognize than that of the other dialects.

We also observe that the combined system performs consistently worse than the dialect-specific systems across all dialects, although the combined system is trained on about 5 times the training data. We obtain the highest increase in WER for Egyptian. Yet in absolute terms, the combined system is still performing relatively well on all the dialects. This may be due to the nature of our data collection: although we asked our volunteers to read the queries as natively as possible, we find that a good percentage of the utterances were read with MSA pronunciations. We speculate that on fully spontaneous data, the difference between the systems will dramatically increase.

It should be noted that there is another advantage of using the dialect-specific systems over the combined system. Our combined system makes use of far more Gaussian mixture components and context-dependent states than the dialect-specific systems. In general, the higher the number of Gaussians and context-dependent states the slower the system is and the more memory it requires. The numbers of Gaussians used in our dialect-specific systems range between 70,000 and 90,000, with about 4,000 context-dependent states, whereas the combined system utilizes about 300,000 Gaussians and about 12,000 states.

4.3. Cross-Dialect Experiments

Next we wanted to know how well an Arabic ASR system trained on one Arabic dialect performs when tested on other dialects.[4] This

[4]This is feasible because all of our data share the same orthography.

System/Test Set	AE	SA	EG	JO	LB
AE	**27.7**	34.4	41.4	22.3	36.8
SA	31.0	**28.7**	41.4	23.0	42.0
EG	37.7	45.9	**24.6**	24.7	36.9
JO	32.6	42.6	38.7	**18.5**	34.8
LB	34.0	44.0	36.9	20.4	**24.2**

Table 4. WER (%) of the Cross-Dialect Evaluation

is particularly important for our decision about the minimal number of systems we need to deploy to cover all the dialects of the Arab World. Since certain queries are significantly more common in one country than others, we decided to perform a cross-dialect acoustic comparison while always employing the target dialect LM and the same undiacritized pronunciation rules. In other words, we use the acoustic models trained on dialect X and test it on the held-out test set of dialect Y, using the LM of dialect Y and a shared lexicon.

Table 4 shows our cross-dialect WER results.[5] As expected, for each dialect test set (columns), the best result is always achieved by the system trained on that dialect. We also observe that all five systems perform relatively well when tested on the Jordanian test set. This is perhaps due to the relatively noise-free Jordanian test set (see Section 4.2). The average within-dialect WER (training and testing on the same dialect) is 24.8%. The cross-dialect WER (training on X and testing on $Y \neq X$) is 35.1%. This large difference suggests that these dialects are in fact acoustically different. We may conclude that at least some of these systems should be built separately.

To further analyze our results, we use the matrix X of cross-dialect WERs from Table 4 to build a symmetric dissimilarity matrix A, where $A_{i,j} = (X_{i,j} + X_{j,i})/2$; $A_{i,i} = 0$. We then visualize our results using Multidimensional Scaling (MDS) to project A into 2D space. Figure 1 shows the data-driven dialect map of our dialects. Interestingly, this map corresponds to the geographical map, except for AE which is located south-east of SA. Note that this map can help us identify what dialects can be combined in one system. We can see, for example, that Egyptian Arabic is quite isolated, which is also supported by linguists – Egyptian Arabic has distinguishable linguistic cues (e.g., syllabic structure is simple).

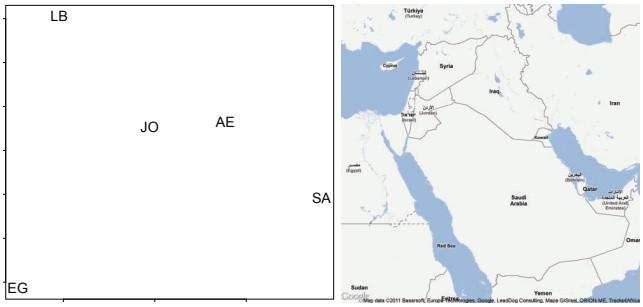

Fig. 1. Data-Driven Dialect Distance Map Derived by MDS from Cross-Dialect WER (left) vs. Political Map (right)

5. CONCLUSIONS AND FUTURE WORK

We have presented Google's Arabic voice search system design. We described our Arabic data collection process from five regional dialects. We found that an Egyptian ASR system that completely ignores diacritics (when building our lexicon and language model) performs better than a system that employs automatically obtained diacritics: short vowels and consonant lengthening are simply modeled as part of the consonant acoustic models. We obtained good WERs (18%–29%) for our five Arabic dialects when each is tested on the same dialect. We also conducted cross-dialect experiments, where we trained on one dialect and tested on the others. When projecting these cross-dialect WERs into 2D space using multidimensional scaling, the resulting dialect similarity map closely corresponds to the geographical map. We conclude that some of our Arabic dialects (e.g., Egyptian) are better handled in a separate dialect-specific system. Finally, we found that dialect-specific systems consistently perform better than a combined system trained on pooled data.

For future work, we plan to revisit all our findings by evaluating on more diverse spontenous test sets. We will experiment with adaptation techniques to support more Arabic dialects using the five collected data sets as our basis.

[5]The diagonal of Table 4 is the same as the first column of Table 3.

6. REFERENCES

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

[2] K. Kirchhoff and D. Vergyri, "Cross-dialectal acoustic data sharing for Arabic speech recognition," in *ICASSP*, 2004.

[3] F. Biadsy, *Automatic Dialect and Accent Recognition and its Application to Speech Recognition*, Ph.D. thesis, Columbia University, 2011.

[4] H. Soltau, L. Mangu, and F. Biadsy, "From Modern Standard Arabic to Levantine ASR: Leveraging GALE for dialects," in *ASRU*, 2011.

[5] T. Hughes, K. Nakajima, L. Ha, A. Vasu, P. Moreno, and M. LeBeau, "Building transcribed speech corpora quickly and cheaply for many languages," in *Interspeech*, 2010.

[6] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature space discriminative training," in *ICASSP*, 2008.

[7] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *9th Intl. Conference on Implementation and Application of Automata*, 2007, `www.openfst.org`.

[8] Y.-H. Sung, M. Jansche, and P. Moreno, "Deploying Google Search by Voice in Cantonese," in *Interspeech*, 2011.

[9] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. ASSP*, 1987.

[10] A. Stolcke, "Entropy-based pruning of backoff language models," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[11] M. Eldawy and M. Taha, "Tashkeel: A library for automatically adding diacritics to undiacritized Arabic text," in *34th Internationalization & Unicode Conference*, 2010.

[12] F. Biadsy, N. Habash, and J. Hirschberg, "Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules," in *NAACL*, 2009.

[13] R. Zbib, S. Matsoukas, R. Schwartz, and John Makhoul:, "Decision trees for lexical smoothing in statistical machine translation," in *Workshop on Statistical Machine Translation and Metrics*, 2010.