# AUTOMATIC PRONUNCIATION PREDICTION FOR TEXT-TO-SPEECH SYNTHESIS OF DIALECTAL ARABIC IN A SPEECH-TO-SPEECH TRANSLATION SYSTEM

*Sankaranarayanan Ananthakrishnan*, *Stavros Tsakalidis*, *Rohit Prasad* and *Prem Natarajan*

Speech, Language and Multimedia Unit
Raytheon BBN Technologies
Cambridge, MA 02138, U.S.A.
{sanantha,stsakali,rprasad,pnataraj}@bbn.com

*Aravind Namandi Vembu*

Ming-Hsieh Dept. of Electrical Eng.
University of Southern California
Los Angeles, CA 90089, U.S.A.
namandiv@usc.edu

## ABSTRACT

Text-to-speech synthesis (TTS) is the final stage in the speech-to-speech (S2S) translation pipeline, producing an audible rendition of translated text in the target language. TTS systems typically rely on a lexicon to look up pronunciations for each word in the input text. This is problematic when the target language is dialectal Arabic, because the statistical machine translation (SMT) system usually produces undiacritized text output. Many words in the latter possess multiple pronunciations; the correct choice must be inferred from context. In this paper, we present a weakly supervised pronunciation prediction approach for undiacritized dialectal Arabic in S2S systems that leverages automatic speech recognition (ASR) to obtain parallel training data for pronunciation prediction. Additionally, we show that incorporating source language features derived from SMT-generated automatic word alignment further improves automatic pronunciation prediction accuracy.

***Index Terms***— speech translation, dialect arabic, pronunciation, speech synthesis

## 1. INTRODUCTION

Modern speech-to-speech (S2S) translation systems are modular in design, typically consisting of three largely independent subcomponents, viz. automatic speech recognition (ASR), statistical machine translation (SMT), and text-to-speech synthesis (TTS) organized in a linear pipeline chain. The ASR and TTS components are both trained on collections of spoken utterances and their corresponding text transcriptions. Both require a lexicon, which maps words to their pronunciations. The SMT system is trained from collections of parallel text, consisting of source language sentences and their corresponding translations in the target language.

In the specific case of dialectal Arabic (e.g. Iraqi), all text transcriptions tend to be *undiacritized*, i.e. words lack diacritics such as short vowels and case endings that serve to disambiguate them. The missing information is inferred by readers based on the context. However, an undiacritized Arabic word by itself may be pronounced and interpreted in more than one way. Thus, an English-to-Arabic (E2A) SMT system trained on undiacritized Arabic text would produce translations with incomplete information. A non-trivial pronunciation prediction step is required to generate a contextually-appropriate phonetic string corresponding to the undiacritized word sequence.

### 1.1. Previous Work

Pronunciation prediction is closely related to the problem of automatic diacritization. Nelken and Shieber [1] proposed a generative "noisy-channel" process that generates undiacritized characters, and used weighted finite-state transducers (WFSTs) to recover the diacritics. Ananthakrishnan et al. [2] used word $n$-gram models to predict the most likely diacritization for a given Arabic word, backing off to character $n$-grams to predict diacritics for unseen words. Zitouni et al. [3] used maximum-entropy models to predict diacritics for each Arabic *letter* using numerous features derived from segment sequences, position of the current letter, part of speech tags, etc. Habash and Rambow [4] train a set of taggers for individual linguistic features (e.g. part of speech, tense, number, gender, etc.), which form the basis of a full morphological tag, and use these to select the best possible diacritization from a set provided by the Buckwalter morphological analyzer. All these approaches have been shown to yield high word-level diacritization accuracy. Since the mapping between diacritized Arabic words and their pronunciations is unique, TTS can use a simple lexicon lookup.

### 1.2. Novel Contributions

Learning models for automatic diacritization requires a hand-diacritized Arabic corpus. While it is available for and has facilitated work on Modern Standard Arabic (MSA), dialectal Arabic lacks comparable resources. This is a major obstacle to developing S2S capability to and from Iraqi Arabic.

Further, previous work on automatic diacritization has been limited to using features derived solely from the Arabic text. However, the S2S framework provides us with additional information in the form of source language features. In an E2A S2S system employing phrase-based SMT, for instance, it is possible to identify the English phrase that generated the current Arabic word for which we are attempting to predict the correct pronunciation. Thus, words from the generating English phrase can be used as additional features to enhance pronunciation prediction accuracy for Iraqi Arabic.

In this paper, we present a weakly supervised approach that leverages a transcribed speech corpus to generate parallel data consisting of undiacritized Iraqi Arabic sentences and their corresponding ground truth pronunciations, alleviating the need for a hand-annotated corpus. We use this corpus in conjunction with SMT word alignments to generate a set of features derived from both target (Iraqi Arabic) and source (English) words for predicting target pronunciations in a maximum-entropy framework.

## 2. GENERATING TRAINING PRONUNCIATIONS

We propose to remedy the absence of a "text-to-pronunciation" corpus for Iraqi Arabic by using ASR forced alignment to automatically choose the correct pronunciation for each word in a transcribed speech corpus. This is far less expensive to create than a manually diacritized Iraqi Arabic corpus.

The acoustic training consisted of 405 hours of Iraqi Arabic speech collected under the DARPA Transtac S2S effort. The audio data span scenarios ranging from checkpoint patrols to medical interviews. Most of these utterances were also manually translated to English. The phonetic pronunciations were obtained from two manually compiled dictionaries provided under the Transtac program, namely the LDC Iraqi Arabic Morphological Lexicon (67K words) and a vowelized dictionary from Appen (65K words). The union of the two phonetic dictionaries contained 103K words. The phoneme set consisted of 53 speech phonemes (47 consonants, 6 vowels), plus silence, garbage and hesitation related phones.

Our ASR training system used a perceptual linear prediction (PLP) front-end that computes 14 cepstral coefficients and normalized energy for each frame of speech. We concatenated 9 contiguous base feature frames resulting in a 135-dimensional feature vector, and then projected it to a 39-dimensional feature space. The feature transformations were based on Linear Discriminant Analysis (LDA), followed by a global Maximum-Likelihood Linear Transform (MLLT). The acoustic model used context-dependent cross-word quinphones with state-clustered tied mixtures and was estimated in the maximum-likelihood (ML) framework using the Baum-Welch algorithm. The pronunciation model was trained over word sequences along with their phonetic sequence.

While the pronunciation of an undiacritized word is not known *a priori*, variants from the lexicon can be used to generate an expanded trellis for forward-backward training of the phoneme HMMs. Since many words in the lexicon have unique pronunciations, the Baum-Welch training algorithm tends to be able to resolve the remaining ambiguities and converge on the correct pronunciation for each word. Thus, acoustic signatures of the different pronunciations in spoken utterances are used to obtain parallel training data for learning pronunciation prediction. With the context-dependent phoneme models trained as described, we obtained the most likely pronunciation sequence for a given spoken utterance and its corresponding undiacritized transcription using forced alignment. Finally, we discarded all utterances for which corresponding English translations did not exist in the parallel SMT training corpus.

## 3. N-GRAM PREDICTION MODEL

The $n$-gram prediction model approximates the joint likelihood of the word sequence $\mathbf{w} = \{w_1 \dots w_N\}$ and the corresponding pronunciation sequence $\mathbf{p} = \{p_1 \dots p_N\}$ as a product of local conditional likelihoods. Equation 1 illustrates this mathematically for a bigram model.

$$p(\mathbf{p}, \mathbf{w}) \approx p(p_1, w_1) \prod_{i=2}^{N} p(p_i, w_i \mid p_{i-1}, w_{i-1}) \qquad (1)$$

Informally, this model assigns the most likely pronunciation to the current word based on its immediate history. The $n$-gram model has been proven to work well in the related automatic diacritization problem; we therefore use it as a baseline system for pronunciation prediction. Note that this model by definition employs only target language features.

In practice, we create a joint corpus consisting of compound tokens formed from the undiacritized Arabic words and their corresponding pronunciations obtained by forced alignment of the speech data. We then use off-the-shelf $n$-gram language model (LM) training tools to estimate a back-off LM from this corpus, using modified Kneser-Ney smoothing.

Inference on a test sentence was performed by creating a confusion network in which each undiacritized input word was paired with all of its possible pronunciations. This network was expanded to a lattice on application of the previously trained $n$-gram prediction LM. Viterbi search on this "scored" lattice yielded the most likely pronunciations for the word sequence.

## 4. MAXIMUM-ENTROPY MODEL

While simple and efficient, the $n$-gram model suffers from two principal disadvantages that limit its performance.

1. It attempts to model the generative process that resulted in undiacritized words and their corresponding pronunciations, rather than directly predicting the latter.
2. It can only incorporate a very limited set of features, viz. the previous $n-1$ word-pronunciation pairs.

The maximum-entropy model [5, 6] alleviates both issues. This model directly predicts the pronunciation for an Arabic word based on a set of input features without jointly modeling the entire sequence. Thus, it does not have to make potentially incorrect Markovian independence assumptions that the $n$-gram model has to. Further, the direct prediction model, unlike the $n$-gram, can use arbitrary features (local and global) for prediction.

Mathematically, the maximum-entropy model estimates the posterior probability of pronunciation $y$ for input word $w_i^t$ based on a set of features derived from the target (Iraqi Arabic) and source (English) word sequences. This is illustrated in Equation 2.

$$p(y \mid \mathbf{w}^t, \mathbf{w}^s) = \frac{exp\left(\sum_{k=1}^{K} \lambda_k f_k(y, \mathbf{w}^t, \mathbf{w}^s)\right)}{\sum_{y \in P_{w_i}} exp\left(\sum_{k=1}^{K} \lambda_k f_k(y, \mathbf{w}^t, \mathbf{w}^s)\right)} \qquad (2)$$

In general, the feature functions $f_k(y, \mathbf{w}^t, \mathbf{w}^s)$ may be derived from the target Iraqi Arabic word sequences $\mathbf{w}^t$ (e.g. surrounding context of the word for which a pronunciation must be predicted) as well as from the corresponding English source words $\mathbf{w}^s$. $P_{w_i}$ denotes the set of valid pronunciations for the current Arabic word $w_i^t$. The weights $\lambda_k$ associated with each feature are estimated to maximize the model's likelihood on the training data. We used the Generalized Iterative Scaling (GIS) algorithm for estimating feature weights. Gaussian prior smoothing was used to alleviate the effects of feature sparsity. The value of this parameter was tuned for minimum prediction error on the development set.

### 4.1. Target Language Features

Following previous work on automatic diacritization, we initially experimented with target language features. In this case, $f_k(y, \mathbf{w}^t, \mathbf{w}^s) = f_k(y, \mathbf{w}^t)$ with reference to Equation 2. Specifically, we considered the following feature sets, which consist of

windows centered around the current word for which a pronunciation must be predicted.

- Current word only: $w_i^t$
- Window of three words: $w_{i-1}^t, w_i^t, w_{i+1}^t$
- Window of five words: $w_{i-2}^t, w_{i-1}^t, w_i^t, w_{i+1}^t, w_{i+2}^t$

Note that each feature is additionally encoded with its position relative to the current word; thus, the same word can trigger a different feature function based on its position.

### 4.2. Source Language Features

The S2S translation framework offers us another set of features derived from the *source* language. Exploiting this additional information source can potentially improve pronunciation prediction accuracy over using target words alone. We augmented the maximum-entropy model with feature functions derived from English source words that generated the target Iraqi Arabic word whose pronunciation must be determined. Due to a complication associated with evaluating pronunciation prediction accuracy in this configuration, we integrated these features within a *translation simulation* framework, as described below.

#### *4.2.1. Translation Simulation*

Incorporating source language features for the current target word is slightly complicated due to our evaluation strategy. In an actual S2S system where the SMT system interfaces with the TTS pronunciation prediction module, we can simply examine the SMT phrase derivations and obtain features from the generating English phrase. However, translating the English source sentences using SMT would give rise to Arabic sentences that may differ significantly from the reference translations. The absence of aligned ground truth reference pronunciations in this scenario would preclude evaluation of the prediction error rate.

We overcame the above issue by *simulating* the translation stage using automatic word alignment in place of SMT phrase derivations. We concatenated the training, development, and validation sets into a single parallel corpus. Unsupervised IBM Model 4 word alignment was performed on this corpus using the publicly available GIZA++ toolkit. To ensure fairness, we assigned unit weight to sentence pairs from the training partition, and zero weight to sentence pairs from the development and validation sets. This minimizes the latter's impact on model parameter estimation, but allows GIZA++ to obtain Viterbi alignments for sentence pairs in these partitions. We performed this alignment procedure in both directions as described in Koehn et al. [7], and combined them to generate a many-to-many (bidirectional) word alignment for phrase translation rule extraction.

Using the above bidirectional word alignment, we extracted phrase translation rules based on heuristics described in Koehn et al. [7]. To generate source language (English) features for a given Arabic word, we scanned the inventory of phrase translation rules extracted from the containing sentence pair. For each target phrase spanning the given Arabic word, we extracted features from English words in the corresponding source phrase. In a very few cases, the rule extraction heuristics produced an inventory in which no target phrase spanned the current Arabic word. In these instances, no source language features were extracted.

| **Partition** | *Train* | *Devel.* | *Valid.* |
|---|---|---|---|
| **Sents.** | 478.7k | 26.6k | 26.6k |
| **Words** | 2.86M | 160.9k | 160.3k |
| **OOVs** | - | 2.4k | 2.4k |

**Table 1**. Summary of Training and Evaluation Corpora

## 5. EXPERIMENTAL RESULTS

We randomly partitioned the force-aligned Iraqi Arabic speech data into training, development, and validation sets as summarized in Table 1. Uniquely, the corresponding English translations are also available for each sentence in all of the above partitions. We used the training partition to estimate $n$-gram and maxent prediction models. The development partition was used for parameter tuning. Finally, trained models and parameters achieving optimal performance on the development set were used to predict Iraqi Arabic pronunciations on the validation set.

### 5.1. Evaluation Methodology

We approached pronunciation prediction as a tagging problem, where each input word is assigned a contextually-appropriate pronunciation. Each assigned pronunciation can be compared to the reference to generate a simple word-level substitution error rate, with appropriate concessions for OOV words. During inference on the development and validation sets with both $n$-gram and maxent models, we assign OOV words a special "unknown" pronunciation, and ignore these items when evaluating and comparing prediction error rate because they impact all competing models and feature sets equally.

Further, only about 14.5k words in a total training vocabulary of 93k words had multiple pronunciations in the ASR training lexicon. The remaining 78.5k words had unique pronunciations, which could be directly looked up in a lexicon. Since these words do not benefit from improved prediction models or enhanced feature sets, we ignored them for scoring purposes. The substitution error rates presented below are therefore based only on Iraqi Arabic words with multiple pronunciations, of which there are 86.1k and 86.5k in the development and validation sets, respectively.

### 5.2. N-gram Prediction

We experimented with 1-, 2-, 3-, and 5-gram LMs. Note that the 1-gram LM is identical to choosing the most likely context-free pronunciation for the current word. This simple yet effective model produced surprisingly high prediction accuracy and was used as a baseline system for comparison. Table 2 summarizes word-level pronunciation prediction performance of various $n$-gram models on the development and validation sets. The accuracy statistics are gathered only with respect to words with multiple observed pronunciations in the main lexicon. The step up from 1-grams to 2-grams improves accuracy by 4.4% relative on the validation set, but further increase in model order degrades performance due to lack of sufficient data to properly capture the generative process.

### 5.3. Maxent Prediction

We trained a separate maxent classification model for each undiacritized word in the training, with feature functions derived from the context as described in Section 4. For each classification model, the

| Model | Development | Validation |
|---|---|---|
| *1-gram* | 36.4% | 36.3% |
| *2-gram* | **34.9%** | **34.7%** |
| *3-gram* | 35.1% | 35.0% |
| *5-gram* | 35.1% | 35.0% |

**Table 2**. Prediction Error Rate for $N$-gram Models

| Model | Development | Validation |
|---|---|---|
| *W* | 36.4% | 36.3% |
| *W+S* | 35.1% | 35.0% |
| *CWC* | 33.8% | 33.9% |
| *CWC+S* | 33.8% | 33.8% |
| *CCWCC* | 33.8% | 33.7% |
| *CCWCC+S* | **33.7%** | **33.5%** |

**Table 3**. Prediction Error Rate for Maxent Models

set of output labels consisted of all pronunciations observed in conjunction with the corresponding input word. To predict the pronunciation for a specific input word, we invoked the appropriate model with the corresponding set of features, and obtained a posterior probability mass function over all possible pronunciations for that word. We chose the output label with the highest posterior probability.

Table 3 summarizes prediction error rates for the maxent model in various feature configurations. In this table, *W* stands for the current Iraqi Arabic word for which a pronunciation must be predicted. Each *C* stands for the neighboring context; for instance, *CWC* refers to the maxent model that derives features from the current, previous, and succeeding words. The *+S* suffix indicates integration of source language features (English words).

As expected, the maxent model that uses only the current word for prediction performs identically to the unigram model. However, there is a relative error reduction of 3.6% upon integrating source features (English words), with $p < 0.001$ according to the NIST Matched Pairs Sentence Segment Word Error (MAPSSWE) significance test. Further improvements are seen when surrounding context is used for pronunciation prediction. The lowest error rate of 33.5% is achieved using the model *CCWCC+S*, which refers to a window of five words centered around the current word in combination with source language features. This outperforms the best target-context-only model *CCWCC* by 0.6% relative ($p < 0.119$), and the best $n$-gram model (2-gram) by 3.5% relative ($p < 0.001$).

## 6. CONCLUSION AND FUTURE WORK

Accurate pronunciation prediction for dialectal Arabic is critical for high-quality TTS in S2S systems. This is hampered by the absence of hand-annotated pronunciation data in addition to lack of rich features beyond the surrounding Arabic context. In this paper, we resolved the annotation problem by using an HMM-based ASR system to force align a transcribed Iraqi Arabic speech corpus in conjunction with a multiple-pronunciation lexicon. The system was able to learn from unambiguous sections of the training data to infer pronunciations for the ambiguous sections. This allowed us to generate a "parallel corpus" of Iraqi Arabic sentences and their ground truth pronunciations from which the mapping could be learnt.

Based on this training corpus, we followed the automatic Arabic diacritization literature in implementing standard $n$-gram and maxent prediction models using target context features. However, the S2S framework allowed us to leverage an additional feature set – namely, the source words that generated the Arabic word in question. In order to preserve the target Arabic words for evaluation, we implemented a simulation that served as a proxy for SMT by leveraging an inventory of phrase translation rules derived from automatic word alignment. Incorporating source (English) words as features within the maxent model based on this word alignment further lowered pronunciation prediction error rate on Iraqi Arabic.

The approach proposed in this paper gave us an objective measure of prediction accuracy for different model types with varying feature sets. Our next goal is to integrate the prediction module within an end-to-end S2S system using actual SMT-generated phrase derivations to obtain source language features. We plan to conduct subjective listening tests in order to determine whether the prediction accuracy improvements due to our innovations actually translate to better speech synthesis quality in terms of intelligibility and fidelity.

## 7. REFERENCES


[1] Rani Nelken and Stuart M. Shieber, "Arabic diacritization using weighted finite-state transducers," in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Stroudsburg, PA, USA, 2005, Semitic '05, pp. 79–86, Association for Computational Linguistics.

[2] Sankaranarayanan Ananthakrishnan, Shrikanth Narayanan, and Srinivas Bangalore, "Automatic Diacritization of Arabic Transcripts for Automatic Speech Recognition," in *Proceedings of the 4th International Conference on Natural Language Processing*, 2005, pp. 47–54.

[3] Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya, "Maximum entropy based restoration of Arabic diacritics," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA, 2006, ACL-44, pp. 577–584, Association for Computational Linguistics.

[4] Nizar Habash and Owen Rambow, "Arabic diacritization through full morphological tagging," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, Stroudsburg, PA, USA, 2007, NAACL-Short '07, pp. 53–56, Association for Computational Linguistics.

[5] Adwait Ratnaparkhi, "A Maximum Entropy Model for Part-Of-Speech Tagging," in *Proceedings of the Empirical Methods in Natural Language Processing*, Eric Brill and Kenneth Church, Eds., 1996, pp. 133–142.

[6] Kamal Nigam, John Lafferty, and Andrew Mccallum, "Using Maximum Entropy for Text Classification," in *IJCAI-99 Workshop on Machine Learning for Information Filtering*, 1999, pp. 61–67.

[7] Philipp Koehn, Franz Josef Och, and Daniel Marcu, "Statistical phrase-based translation," in *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Morristown, NJ, USA, 2003, pp. 48–54, Association for Computational Linguistics.