A Weighted Combination of Speech with Text-based Models for Arabic Diacritization

Aisha S. Azim, Xiaoxuan Wang, Khe Chai Sim

School of Computing, National University of Singapore, 13 Computing Drive, Singapore
aisha@comp.nus.edu.sg, wangxx@comp.nus.edu.sg, simkc@comp.nus.edu.sg

Abstract

The majority of studies on Arabic diacritization have employed textually inferred features alone. This paper proposes a novel approach, where the weighted combination of speech with a text-based model is used to allow linguistically-insensitive acoustic information to correct and complement the errors generated by the text model’s diacritic predictions. The acoustic model is based on Hidden Markov Models and the textual model on Conditional Random Fields. The combination brings significant reduction in error rates across all metrics, especially in case endings, which are the most difficult to predict. It gives results superior to those of conventional methods, with diacritic and word error rates of 1.5 and 4.9 inclusive of case endings, and 1.0 and 2.7 exclusive of them.

Index Terms: Arabic diacritization, case endings, multimodal systems

1. Introduction

The Arabic alphabet consists of 28 consonants and 6 short vowels. The Arabic script is written as a sequence of consonants, without vowels or other pronunciation cues. Given the language’s highly inflective nature and morphological complexity, this means that a single sequence of consonants could easily represent over fifty unique words, because of the several unique sequences of diacritics that could be applied to it. While this is not problematic for native readers, it poses serious problems in textual disambiguation for automated systems. Hence diacritization becomes a necessary basic step for most downstream processes.

The eight diacritics are described in detail in [1]. Inflectional diacritics, or case endings, typically appear on the final consonant of a word. These are the most difficult to predict since they are dependent on deep syntactic information. They play a significant role in increasing a system’s Word Error Rate (WER) and Diacritic Error Rate (DER).

Many approaches have been proposed to deal with the diacritization problem [2], [3], [4], typically drawing information from textually extracted features alone. Acoustic input has only been considered in the context of automatic speech recognition (ASR), for example in [5], where the authors show that textual features are required in addition to a low-performance speech recognizer. But diacritization has never addressed speech based on the merits of acoustic information itself.

In this paper we explore the extent to which acoustic input improves diacritization, using a weighted combination of two independent systems, one modeling speech and the other modeling text.

2. Problem formulation

BAMA\(^1\), the Buckwalter Arabic Morphological Analyzer, has been used extensively for Arabic diacritization, because it can generate a list of candidate diacritized solutions for a given input word. Hence predicting the correct diacritics for a word can be treated as a selection from this list.

The system in this paper accepts two streams of data: raw Arabic text, and an acoustic signal of the correctly vocalized speech corresponding to that text. Two independent diacritizers are employed, one that is text-based and modeled by Conditional Random Fields (CRF); and one that is speech-based and modeled by Hidden Markov Models (HMM). Fig. 1 summarizes the system’s architecture and process, given an input word \(w\).

Let \(W\) be the set of raw, undiacritized input words. Then for each word \(w \in W\), we have a set of potential diacritized solutions, \(D_w\). For this system, the solutions were generated by BAMA 2.0. Since our task is diacritization, we are only concerned with the unique sequence of diacritics on a string of consonants, and not with its morphological analyses. Therefore if BAMA produces several different morphological possibilities for a word, all with the same sequence of diacritics, then the word is counted as having a single solution.

If a word \(w\) has \(n\) unique solutions, then \(D_w\) is expressed as:

\[
D_w = \{d_{w,1}, d_{w,2}, \ldots, d_{w,n}\}
\]  

(1)

This set of solutions is taken as input to both diacritizers. Let \(x\) be a tuple \((d_{w,i}, s_{w,i})\) that relates a potential solution \(d_{w,i} \in D_w\), with its likelihood score, \(s_{w,i}\). The

\(^1\)http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004L02
acoustic and text models are used to independently generate each solution’s score, producing a set of \( n \) tuples for each word. The scoring is discussed in Section 3.

We denote the speech-based diacritizer’s scored set of solutions for a word \( w \) by \( S_w \), and the text-based diacritizer’s by \( T_w \). \( S_w \) can be described as:

\[
S_w = \{ x | x = (d_{w, 1}, s_1), d_{w, 1} \in D_w, w \in W \} \quad (2)
\]

Similarly, \( T_w \), with likelihood score \( t_{w, i} \), is:

\[
T_w = \{ x | x = (d_{w, i}, t_{w, i}), d_{w, i} \in D_w, w \in W \} \quad (3)
\]

Let the score of a tuple in these sets be denoted by \( S_w[d_{w, 1}] \) and \( T_w[d_{w, i}] \). A weighted combination is applied to the tuples in sets \( S_w \) and \( T_w \), using weights \( \alpha \) and \( \beta \), and the final diacritized solution \( d_w^* \) is that which maximizes the combination. This is expressed below:

\[
d_w^* = \arg \max_{d_{w, i} \in D_w} (\alpha \cdot T_w[d_{w, i}] + \beta \cdot S_w[d_{w, i}]) \quad (4)
\]

where \( \alpha + \beta = 1 \).

3. System description

3.1. Speech-based diacritizer

The acoustic model was trained on approximately 2.5 hours of spoken utterances, read out from selections of the 510 Al-Nahar articles in the training set described in [1], taken from the ATB3 corpus [6]. The speech was recorded by one speaker, in a clean environment, using the reference diacritization given by human annotation. The model was built using flat-start, 3-state HMMs with 26 Gaussian components per state, and 52-dimensional MFCC feature vectors [7]. Each of the 28 Arabic consonants, its geminated variant, the two diphthongs, three vowels and the pharyngealized \( L \) were modeled. The phones were converted into tied-state context-dependent triphones, using decision tree clustering based on the classification of Arabic consonants described in [8].

Once the model is trained, then speech-based diacritization for a given word \( w \) is a four-step process:

1. Obtain all diacritization possibilities. From the solutions generated by BAMA for \( w \), we select all unique solutions. The ATB3 corpus provides these solutions.

2. Convert solutions to phonetic transcriptions. We use a rule-based grapheme-to-phoneme (G2P) layer to convert each potential solution \( d_{w, i} \in D_w \) into phonetic transcriptions compatible with the speech recognizer. The G2P rules deal with the normalization of the letters Alif and Alif Mak-soorah, Taa Marbootah, Hamza, and the solar and lunar letters. More information on G2P rules can be found in [8].

3. Force alignment to obtain word boundaries. Using the acoustic model, we apply a forced alignment between the speech input and the phonetic transcriptions of the reference data, obtained from our G2P layer above. This is done in order to identify \( w \) in the speech input. Given an undiacritized word \( w \), its corresponding utterance is identified in the speech input by extracting its word boundaries in time, generated during alignment.

4. Evaluate maximum likelihood score. We run another force-alignment, this time between the acoustic signal corresponding to \( w \) (Step (3)) and the phonetic transcription of each solution \( d_{w, i} \) (Step (2)). From this alignment, the speech recognizer assigns an acoustic log likelihood score to \( d_{w, i} \) using the Viterbi algorithm. The acoustic scores are normalized for the utterance of each word \( w \), and the solution picked by the speech-based diacritizer, \( d_w^* \), is the solution with the maximum log likelihood:

\[
d_w^* = \arg \max_{d_{w, i} \in D_w} [S_w] \quad (5)
\]

3.2. Text-based diacritizer

The text model is built using CRFs [9]. CRFs are powerful at capturing context because they relax the independence assumption inherent in HMMs, and hence are ca-
able of defining conditional probabilities between labels and observations. As in [4], input text is normalized so that there is a one-to-one mapping between observations and labels. The following features are used: a training window of 9 consonants (i.e. the current consonant, four consonants to the left and four to the right); the current, previous and next words; and the part-of-speech (POS) tags for the current, previous, and next words. The model also incorporates a vowel bigram. Let this set of features be \( f \). The text model was trained on approximately 470K words extracted from Al-Nahar articles, available in the ATB3 corpus and Arabic Gigaword Fourth Edition\(^2\). The Arabic Gigaword data was diacritized using MADA [3].

For text-based diacritization, a word \( w \) is input as a sequence of raw Arabic consonants, \( C \). Each consonant \( c \in C \) may be assigned one of fifteen possible labels from the set \( V \); this set comprises all valid diacritics that may appear on a consonant\(^3\). For each label \( v \in V \) per consonant \( c \in C \), the marginal probability \( p(v|f) \) may be computed.

Given the per-label marginal probabilities, provided by CRF++\(^4\), we use the CRF model to compute how likely a particular solution \( d_{w,i} \) is. Let \( V_{w,i} \subset V \) be the sequence of labels proposed by \( d_{w,i} \) as the solution. For example, if the input word is the string of consonants \( \text{whm} \), and if a possible solution is \( \text{waomi} \), then the sequence of labels \( V_{w,i} \) in this case is \( \text{a} \). We calculate the score of solution \( d_{w,i} \) by summing over the logs of the marginal probabilities associated with its labels:

\[
T_w[d_{w,i}] = \sum_{v \in V_{w,i}} \log p(v|f) \tag{6}
\]

The best diacritized solution in the text-based context, \( d_{w,*}^t \) is:

\[
d_{w,*}^t = \arg \max_{d_{w,i} \in D_w} [T_w] \tag{7}
\]

### 4. Weighted linear interpolations: evaluation and experiments

#### 4.1. Combination

Text and speech cannot be directly combined at the same granular level. While text can be normalized into consonant-vowel pairs for processing, the same cannot be done with the acoustic model’s phonetic transcriptions. For example, the word “phone” in English can be split it into the pairs: \((p, \text{no-vowel}), (h, o), (n, e)\). However there is no direct way to map these pairs to the phonetic transcription: /f oh n/. Therefore we operate on word-level log likelihoods produced by the speech recognizer, as opposed to diacritic-level scoring by the text-based system (see Equation (6)). Since the acoustic scores are in the log domain, logs are taken of the text-based scores. \( T_w \) and \( S_w \) are now compatible for interpolation.

Table 1 presents an example of tuples from sets \( T_{10} \) and \( S_{10} \). The set of solutions for word 10 is \( D_{10} \). Each solution proposes a sequence of vowels corresponding to the word’s consonants, \( \text{whm} \). Solutions 6 and 9 are shown. Solution 6 proposes the vowel sequence \( \text{aoi} \), and is stored with its text-based score to form the first tuple, \((d_{10,6}, t_{10,6}) \in T_{10} \) in Row 1, and with its acoustic score to form the second tuple, \((d_{10,6}, s_{10,6}) \in S_{10} \) in Row 2.

<table>
<thead>
<tr>
<th>Candidate Tuples</th>
<th>Consonants</th>
<th>Vowels</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ((d_{10,6}, t_{10,6}) \in T_{10})</td>
<td>\text{whm}</td>
<td>\text{aoi}</td>
<td>-19.9</td>
</tr>
<tr>
<td>2. ((d_{10,6}, s_{10,6}) \in S_{10})</td>
<td>\text{whm}</td>
<td>\text{aoi}</td>
<td>-52.5</td>
</tr>
<tr>
<td>3. ((d_{10,9}, t_{10,9}) \in T_{10})</td>
<td>\text{whm}</td>
<td>\text{au}</td>
<td>-0.01</td>
</tr>
<tr>
<td>4. ((d_{10,9}, s_{10,9}) \in S_{10})</td>
<td>\text{whm}</td>
<td>\text{au}</td>
<td>-53.2</td>
</tr>
</tbody>
</table>

Table 1: Candidate tuples and their scores

When we ran the system, the text-based diacritizer initially selected Solution 6, but after the interpolation with acoustically informed solutions, the best combined score selected was that of Solution 9. Solution 9 was indeed the correct choice.

#### 4.2. Evaluation

The experiments in this section are tested\(^5\) using the ATB3 datasets [6] that have been accepted widely in the literature [1]. From the established training dataset of approximately 52K words, we set aside 1465 of them as development data\(^6\). The evaluation metrics used are the same as those in the literature, with two points to note.

Firstly, current literature uses all tokens from the established datasets in calculating error, but we exclude numbers and punctuation. This is because there is no variation in learning these tokens’ diacritics; they all have “no-diacritic”, hence including them in calculations portrays a slightly optimistic measure of true diacritization accuracy. Secondly, in the case of computing error without case endings (\(\text{no CE}\)), the final consonant of a word has also conventionally been included in the calculations.

We suggest a new metric: \(\text{DER}_{\text{abs no CE}}\) - this is the absolute \(\text{DER}\) that excludes the final consonant in calculations. We believe this more accurately brings out the difference between a system’s diacritization performance that includes case endings, and that which does not.

Figure 2 describes the weighted interpolations on the development dataset. The interpolations are represented as the ratio of the text to speech (\(T:S\)). The optimal values for the weights \( T \) and \( S \) were found to be 0.2 and 0.8.

\(^2\)http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2009T30

\(^3\)In Buckwalter transliteration, these labels are: \(a, u, i, o, F, N, K, ~a, ~u, ~i, ~o, ~F, ~N, ~K, \text{and no diacritic} \), denoted by \(\epsilon\).

\(^4\)http://crfpp.sourceforge.net/

\(^5\)Approx. 200 sentences were excluded from the test dataset, as they contained words with no solutions in the corpus. This left us with 80% of the data, still enough for a reasonable comparison with other systems.

\(^6\)The development dataset comprised 77 sentences between 20020115_0001 and 20020115_0007 (approx. 25 minutes of speech).
These weights were then used on the test data. Table 2 reports the results. The error rates of the individual diacritizers are listed under Text and Speech. Within the Text column, DER$_{abs}$ differs from DER CE by 46.5%. However, the difference between these two metrics in the Speech column is significantly lower. This confirms the hypothesis that text features contribute to systematic inflictional errors, while acoustic features generate more regular errors throughout the text. To report the error in predicting case endings only, which appear on word-final consonants, we add an additional row to the table, DER CE only. Note that while the text-based system incorrectly predicted 13.2% of all case endings, the speech-based diacritizer only erred in 6.2%. DER$_{abs}$ NO CE represents non-CE diacritization, which excludes all word-final consonants.

### Table 2: Error rates (%) on test data

<table>
<thead>
<tr>
<th>Metric</th>
<th>Text</th>
<th>Speech</th>
<th>Speech+Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>DER CE</td>
<td>4.3</td>
<td>3.2</td>
<td>1.5</td>
</tr>
<tr>
<td>DER$_{abs}$ no CE</td>
<td>2.0</td>
<td>2.5</td>
<td>1.2</td>
</tr>
<tr>
<td>DER no CE</td>
<td>1.6</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>WER CE</td>
<td>16.5</td>
<td>9.6</td>
<td>4.9</td>
</tr>
<tr>
<td>WER no CE</td>
<td>5.0</td>
<td>5.3</td>
<td>2.7</td>
</tr>
<tr>
<td>DER CE only</td>
<td>13.2</td>
<td>6.2</td>
<td>2.8</td>
</tr>
</tbody>
</table>

5. Discussion

Previous work in the field [5] suggested that starting with a speech-based diacritizer, it was necessary to incorporate textual features for quality results. This paper takes a related but different stand. It demonstrates that using two independent diacritizers, acoustic information can be used as an important component of a system to correct and complement the errors produced by the text-based component, and that its contribution to the prediction of case endings is significant. Table 3 below summarizes the error rates of two state-of-the-art systems alongside the error rates of the combined system (Speech+Text) presented in this paper. The systems presented are MADA

6. Conclusion

This paper presented a novel system that combined speech with text to generate diacritization results that are superior to current state-of-the-art systems. Acoustic information and textual information were found to correct and complement each other, while case-ending errors produced by the text-diacritizer were significantly reduced. The system is useful for STT and language learning applications.

7. References


Table 3: Error rates (%) for state-of-the-art systems

<table>
<thead>
<tr>
<th>Metric</th>
<th>MADA</th>
<th>SHD</th>
<th>Speech+Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>DER CE</td>
<td>4.8</td>
<td>3.8</td>
<td>1.5</td>
</tr>
<tr>
<td>DER no CE</td>
<td>2.2</td>
<td>1.2</td>
<td>1.0</td>
</tr>
<tr>
<td>WER CE</td>
<td>14.9</td>
<td>12.5</td>
<td>4.9</td>
</tr>
<tr>
<td>WER no CE</td>
<td>5.5</td>
<td>3.1</td>
<td>2.7</td>
</tr>
</tbody>
</table>