This is a repository copy of *Diacritization of a Highly Cited Text: A Classical Arabic Book as a Case*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/128591/

Version: Accepted Version

**Proceedings Paper:**

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

**Reuse**
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

**Takedown**
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.
Diacritization of a Highly Cited Text: 
A Classical Arabic Book as a Case

Abdulrahman Alosaimy 
School of Computing 
University of Leeds 
Leeds, UK 
scama@leeds.ac.uk

Eric Atwell 
School of Computing 
University of Leeds 
Leeds, UK 
e.s.atwell@leeds.ac.uk

Abstract— We present a robust and accurate diacritization method of highly cited texts by automatically “borrowing” diacritization from similar contexts. This method of diacritization has been tested on diacritizing one book: “Riyad As-Salheen”, for the purpose of morphological annotation of the Sunnah Arabic Corpus. The original source of Riyad is about 48.66% diacritized, and after borrowing diacritization, the percentage jumps to 76.41% with low diacritic error rate (0.004), compared to 61.73% (DER=0.214) using MADAMIRA toolkit, and 67.68% (DER=0.006) using Farasa toolkit. More importantly, this method has reduced the word ambiguity from 4.83 diacritized form/word to 1.91.

Keywords—diacritization; Arabic; NLP; Sunnah; Riyad As-Salheen

I. INTRODUCTION

In the Arabic language, a high amount of phonological information is missing such as short vowels, Shaddah, tanween, Maddah, and sometimes hamzah1 as well. They (collectively called diacritics) are not usually written. As a result, the ambiguity at the word level is high in Arabic. There is an average of 11.5 diacritizations/word according to [1]. For example, a vowelized form of the word 

1 In cases where Hamza is considered a diacritic, only different shapes of Hamza on Alif is considered.
Diacritization is usually done fully, but this full diacritization is not necessary diacritizing each letter, due to the missing standard definition of the fully-diacritized word. There are some letters that are not diacritized even in lexicons, and by convention are no-vowel letters (i.e. has an intuitive vowel but not written). For example, using some diacritization standards, the letter that precedes a long vowel and the lam letter in definite AL article are two no-vowel letters. However, deciding whether Waw/Yaa letters are consonant or a vowel is ambiguous. Similarly, deciding whether the lam is part of a definite AL article is ambiguous too.

Arabic diacritization has grabbed the attention of Arabic NLP researchers, and much work has been done. Previous approaches have focused on improving the quality of automatic diacritization to produce a fully diacritized version of the text, either using rule-based approach [6], statistical approaches using, for example, recurrent networks [7], n-gram model [8], or hybrid approaches which usually perform the best [9]–[11]. This work, however, focuses on diacritizing text for the purpose of manual annotation later. That is, the diacritization approach seeks a high accuracy in diacritization but is not necessary to diacritize the full text. This approach crosses some interests with [4] which exploits diacritizing to improve morphological annotation. Our methodology is unique as it exploits partial diacritized texts as a source for diacritization. We borrow partial diacritizations from similar contexts and merge them together, and hope it lowers the ambiguity level of that word as much as possible.

II. MOTIVATION

This article is motivated by our project of developing semi-automatically annotated Sunnah Arabic Corpus. Since its text is not been fully diacritized, we needed to adopt a method for diacritizing. Since the corpus mostly consists of texts that are highly quoted in other diacritized texts, we had the idea of “borrowing” their diacritization.

In many Classical Arabic texts, it is common to diacritize the word at least minimally: to the amount that is enough to remove the ambiguity to the readers. However, this borderline is not clear enough, and words that seem clear to the writer might still be ambiguous to a reader. Therefore, we notice different diacritization of the same word in different positions within the book, or between different versions of the book. These differences are exploited for the sake of improving morphological annotation of our Sunnah Arabic Corpus (SAC) and reach the minimal diacritization for each word.

III. DATA

We picked one book from our Sunnah Arabic Corpus: Riyāḍu Aṣṣāliḥīn2 (aka The Meadows of the Righteous) which is a compilation of 1896 hadith narratives written by Al-Nawawi and published in 1334. The total number of words in Riyad is around ~144k (~17k word types), and 48.66% of its letters are diacritized. Riyad was chosen due to several reasons:

1. It compiles narrations reported in other Hadith books (e.g. Albukhari) which make them a good source for diacritization.

2. Its codex was validated and investigated by several scholars by scientific palaeographical process; at least there are two digitally available validated versions of the same text.

3. Its narratives have been explained in 6 written books.

The currently available diacritized corpora are either annotated corpora (mainly news) and Tashkeela (religious texts) [12], a corpus of 6.15 million words, which we used as an initial source for diacritization. But since Tashkeela focuses on fully diacritized texts, we added several Hadith books downloaded from Shamela library. Shamela (http://shamela.ws) is a downloadable library that contains at least 5300 Arabic books in Islamic studies and becomes the standard library of Arabic classical books. It has been used to obtain Arabic classical text in building several corpora [12]–[14].

Although having a large collection should not lower the accuracy of our method, we limit the corpus size in our experiments for training time efficiency. We picked relevant books from Tashkeela and

---

2 All experiments in this paper and their used data is available at: http://github.com/aosaimy/sac
Shamela, i.e. books that have a high likelihood of quoting texts from Riyad. This selection method is done manually. This selection can be done automatically as we developed a small companion tool that measures one book’s contribution by computing the number of matching n-grams with additional diacritization. The final corpus is 7677814 words, where 58.31% of its letters are diacritized.

In Arabic examples for the rest of this article, we use Buckwalter transliteration\(^3\) instead, as it is easier to examine the differences of the diacritics. Diacritics are small glyphs, and the differences might not easily visually noticeable. Please note that these diacritics does not represent the possible diacritization states of one letter as they can be combined (especially with Dhammah), and some letters only accept a subset of them. The maximum total number of states is 18.

\[
\begin{array}{c|c|c|c}
F & \text{Fatha Tanween} & N & \text{Dhammah Tanween} \\
K & \text{Kasrah Tanween} & a & \text{Fatha} \\
u & \text{Dhammah} & i & \text{Kasrah} \\
\sim & \text{Shaddah} & o & \text{Sokun} \\
\end{array}
\]

IV. METHODOLOGY

Since the text in Riyad is highly cited and quoted, we have increased its text diacritization level, by automatically “borrowing” diacritization from other books. We developed an open-source diacritizer\(^4\) that matches undiacritized version of one word in Riyad with its equivalent in other books using their word n-gram concordance. Algorithm 1 describe formally the method which could be explained in more details as follows:

1. It converts target text into a list of word n-grams, with reference to its locations in text, diacritized and undiacritized versions of the centre word.
2. It reads documents in source corpora in parallel. For each n-gram that is on our list (after normalization), it builds a list of matching word-ngrams.
3. For matching n-grams, it extracts variant diacriticizations of the centre word and counts the number of occurrences of that diacritization.

4. Once finished, variants are sorted by the number of occurrences to prevent infrequent diacritization from bubbling up to the surface diacritization in the next step.

5. Centre words variants are merged recursively: The merge procedure (Algorithm 2) is done letter by letter, and for every letter, only candidate diacritics that do not contradict with one existing are merged.

6. (extended version) uses morphological analyser (MA) to improve the results if

---

\(^3\)available freely at http://github.com/aosaimy/arabic-vowelizer

---

\(^4\)http://www.qamus.org/transliteration.htm
Diacritics Standardization

Word diacritization is the same for n surrounding

grams that has variant
diacritization of its centre word. Then, we examine
the top of the list (top 100), ranked based on the
number of variants in descending order. Table IV lists
a sample of top n-grams of first experiments for n=5.

All variants did not show a sign of artificial
diacritic, nor show a non-standard diacritization. The
centre word has no conflicting diacritization for 98%
of the top 100 of the list. Conflicting diacritization is
due to different pronunciation of proper nouns,
misspelt diacritics, or improper last diacritic.

D. Similarity between source and target corpora

SAC is mostly a collection of religious text which
is widely quoted. Its content has been explained by
several authors. This increases the chance that its text
has been quoted. The results of our experiment show
that at least 84.34% of the corpus word n-grams has
been found in the source corpora.

E. There is no other diacritized form if
morphological analyser says so

We used four morphological analysers to increase
the diacritization coverage for our corpus. By
merging the output of analysing each word, we built
a list of possible diacritization of each word. After
close examination of the results, their level of
diacritization is different. The diacritized format is
not usually full. Table II showed the diacritization
coverage for each analyser. While merging analysers'
results increases the coverage, similar words do not
merge as their level of diacritization is different;
which results in having more than one form of
diacritization when in fact there should be one. This
explains the jump in the number of possible
diacritization from 10.38 (at maximum) to 17.42.

Using SAWAREF toolkit [15], we run four
morphological analysers, namely Elixir Functional
Morphology (EX) [16], ALMORGEANA (included
in MADA toolkit) (AL) [17], AraMorph (BP) [18],
and ALKhalil (KH) [19], on the lexicon of Riyadh
Asslaheen (17600 distinct words). The average
number of possible diacritized forms is shown in
Table I.

<table>
<thead>
<tr>
<th>MA</th>
<th>Max</th>
<th>Mean</th>
<th>Median</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX</td>
<td>124</td>
<td>8.46</td>
<td>6</td>
<td>67.46%</td>
</tr>
<tr>
<td>KH</td>
<td>96</td>
<td>10.38</td>
<td>7</td>
<td>80.64%</td>
</tr>
<tr>
<td>BP</td>
<td>20</td>
<td>2.38</td>
<td>2</td>
<td>47.67%</td>
</tr>
<tr>
<td>AL</td>
<td>23</td>
<td>3.69</td>
<td>3</td>
<td>42.65%</td>
</tr>
</tbody>
</table>
We only use MA diacritization if it matches only one form. Using a random sample (of 100 words) that matches this criterion, we could not spot a single error in the enhanced diacritization. This suggests that it is safe to assume there is no other diacritized form if morphological analyser says so.

V. Evaluation

Our evaluation uses two metrics for accuracy, and coverage, both in terms of character level. Accuracy is measured by Diacritic Error Rate (DER), i.e. a letter is marked correctly if it has all diacritics in the original text. Coverage is measured by the percentage of letters that has at least one diacritic.

In addition, we introduce ambiguity measure defined as the practical average of the possible number of diacritizations per word. In theory, if a word of three letters has no diacritics, there are at least eight possible diacritization for each letter (final letter can have more). But we report the practical number of diacritizations only, extracted from a lexicon (or in our case morphological analysers). In case a partially diacritized word, the morphological analyser will only return the subset of possible diacritizations that has match given diacritization.

\[
DER = \frac{\sum f(w_i)}{\sum |t(w_i)|} \\
f(w) = |s|, s \subset v(w) \text{ that is incorrect} \\
Coverage = \frac{\sum_{|v_i|\geq 0} |v_i|}{\sum |t(w_i)|} \\
\text{Ambiguity} = \frac{\sum_{|w|} \text{ambig}(w_i)}{|w|} \\
\text{ambig}(w) = |MA(v(w))|
\]

We test on the part of the text that is already diacritized. In other words, we used our models to diacritize a completely undiacritized version of Riyadh, and later test the accuracy and coverage of our assumption on the diacritized version. However, since this method does not diacritize the full text, we only evaluate the letters with a diacritic.

In Table II, we compare the accuracy (in terms of DER), coverage, and the word ambiguity after diacritization of six models of diacritization. We can see that accuracy improves when word’s context is larger, but on the other hand, the coverage drops. Word ambiguity does not change after using MA, as MA’s diacritization is not used unless word diacritization only matches one candidate. The accuracy increased very slightly (about 0.0001) when using MA; however, the coverage increased by ~0.2.

<table>
<thead>
<tr>
<th>Model</th>
<th>Coverage</th>
<th>DER</th>
<th>Ambiguity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undiacritized</td>
<td>0</td>
<td>N/A</td>
<td>17.42</td>
</tr>
<tr>
<td>Baseline</td>
<td>48.66%</td>
<td>N/A</td>
<td>4.83</td>
</tr>
<tr>
<td>3-gram</td>
<td>80.32%</td>
<td>0.007</td>
<td>1.56</td>
</tr>
<tr>
<td>3-gram+MA</td>
<td>81.26%</td>
<td>0.007</td>
<td>1.56</td>
</tr>
<tr>
<td>5-gram</td>
<td>76.41%</td>
<td>0.004</td>
<td>1.91</td>
</tr>
<tr>
<td>5-gram+MA</td>
<td>77.70%</td>
<td>0.004</td>
<td>1.91</td>
</tr>
<tr>
<td>7-gram</td>
<td>73.97%</td>
<td>0.003</td>
<td>2.13</td>
</tr>
<tr>
<td>7-gram+MA</td>
<td>75.59%</td>
<td>0.003</td>
<td>2.13</td>
</tr>
</tbody>
</table>

Additionally, we compare our results to two major available diacritizers: MADAMIRA [10] and FARASA [9]. Diacritization is normalized for both toolkits. Our 5-gram model slightly surpasses both tools, and FARASA scored an error rate is 0.006 while MADAMIRA was not performing well: 0.214, which is due to the fact that MADAMIRA removes original diacritics before processing the text. For a fair comparison, we re-compute the error rate given undiacritized version; FARASA error rate jumped to 0.263, and the DER of our 5-gram model increased slightly to 0.008.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Coverage</th>
<th>DER</th>
<th>Input Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>MADAMIRA</td>
<td>N/A</td>
<td>N/A</td>
<td>Diacritized</td>
</tr>
<tr>
<td></td>
<td>61.73%</td>
<td>0.214</td>
<td>Undiacritized</td>
</tr>
<tr>
<td>FARASA</td>
<td>67.68%</td>
<td>0.006</td>
<td>Diacritized</td>
</tr>
<tr>
<td></td>
<td>65.36%</td>
<td>0.263</td>
<td>Undiacritized</td>
</tr>
<tr>
<td>5-gram</td>
<td>76.41%</td>
<td>0.004</td>
<td>Diacritized</td>
</tr>
<tr>
<td></td>
<td>71.81%</td>
<td>0.008</td>
<td>Undiacritized</td>
</tr>
</tbody>
</table>

While the two tools are expected to diacritize the text thoroughly, we found that MADAMIRA only diacritized 61.73% of letters, and FARASA only diacritized 65.36%, 67.68% for undiacritized, and diacritized input text respectively. Using our method, the 5-gram model diacritized 71.81% of letters. This is due to diacritization standards of final letter, article AL and long vowels in addition to the fact that our measure does not tolerate letters with obvious diacritics (such as Alif Madd (١), Alif (١) and Lower Hamza (١)). Even the Quran text (extracted from
Tanzil Project), which is known to have a full diacritized form, covers only 77.83% of letters. Table III summarizes these findings.

VI. CONCLUSION

We presented and evaluated a methodology for diacritizing highly quoted texts by borrowing diacritization from its citations. This method exploits and reuses manual diacritization from other works. To fully exploit diacritized text from different origins, we had to deal with diverse diacritization standardization. This method is unique in reusing partially diacritized text as a source for diacritization.

By matching the undiacritized version of one word in target text with its equivalent standardized version in other books using their word n-gram concordance, the percentage of diacritized words in Riyad As-Salheen rose with high accuracy. We compared different models of our method intrinsically, and extrinsically with available external diacritizers.

We urge linguists and researchers to develop standards way of diacritization. We plan to extend this work to build a fully diacritized corpus of highly quoted texts. We plan to incorporate this method with our morphological annotation tool (Wasim)[21], as a helper for diacritization annotation layer. Additionally, merging diacritized forms from morphological analysers could help to build a more robust ensemble tagger. Finally, we plan to test the possibility of applying same method to recover missing Hamzah.

A. References


