Higher Order n-gram Language Models for Arabic Diacritics Restoration

Yasser Hifny
University of Helwan, Egypt
yhifny@fci.helwan.edu.eg

Abstract
Dynamic programming based Arabic diacritics restoration aims to assign diacritics to Arabic words. The technique is purely statistical approach and depends only on an Arabic corpus annotated with diacritics. The possible word sequences with diacritics are assigned scores using statistical n-gram language modeling approach. Using the assigned scores, it is possible to search the most likely sequence using a dynamic programming algorithm. In previous work [1], the assigned scores are based on a bigram stochastic language model and the decoder was restricted to this model. Using higher order n-gram language may lead to better diacritization accuracy. In this work, we extend the dynamic programming decoding algorithm to support higher order language models. Preliminary results on a public domain corpus show that dynamic programming decoding based on higher order n-gram models can lead to better results than bigram models.

Index Terms: Arabic diacritics restoration, dynamic programming, statistical language modeling, smoothing

1 INTRODUCTION

Arabic diacritics restoration aims to assign diacritics to Arabic words. Over the last few years, many systems to restore Arabic diacritics were developed [2],[3],[4],[5],[6],[7],[8]. When the diacritics are present, the Arabic script provides enough information about the correct pronunciation and the meaning of the words. In some applications like Arabic text to speech, diacritics are necessary to get the correct pronunciation [9]. Moreover, diacritics help to get the reference transcription for speech recognition systems [10].

In [1], an algorithm to restore Arabic diacritics using a dynamic programming approach was presented. The possible word sequences with diacritics are assigned scores using statistical n-gram language modeling approach. Using the assigned scores, it is possible to search the most likely sequence using a dynamic programming algorithm. The described algorithm is closely related to the Viterbi algorithm which is used to find the most likely sequence in the hidden Markov models. However, in our formulation we do not have hidden states as in Hidden Markov Models (HMMs). The presented technique is purely statistical approach and depends only on an Arabic corpus annotated with diacritics.

In previous work, the assigned scores are based on a bigram stochastic language model and the decoder was restricted to this model [1]. Using higher order n-gram language models may lead to better diacritization accuracy. In this work, we extend the dynamic programming decoding algorithm to support higher order language models. Higher order language models can be easily estimated using a language modeling toolkit. However, the search algorithm based on static lattices presented in [1] cannot be used to decode higher order n-gram language models. The new search algorithm will depend on dynamic lattices where the scores of different paths will be computed on the run time. Hence, the arcs scores can depend on the decoded history and higher order n-gram language models can be supported.

In Section 2, a mathematical formulation of the restoration of Arabic diacritics is described. The new search algorithm based on dynamic lattices is presented in 3. Sections 4 and 5 give experimental results on a public domain task and conclusions.

2 PROBLEM FORMULATION
A fundamental problem in natural language processing is to give a score to a sentence hypothesis or sequence of words. The assigned score for a sentence can be used to disambiguate between different solutions. Large vocabulary speech recognition systems (LVCSR) use a n-gram language model, which gives an approximate probability score of an
allowable word sequence $W = w_0 w_1 \ldots w_m$ in the recognition task. This probability score is calculated by accumulating local scores using $n - 1$ Markov chains over the word sequence and is given by

$$P(W) = \prod_{i=1}^{m} P(w_i|w_{i-n+1})$$  \(1\)

For bigram language model, this probability score is given by

$$P(W) = \prod_{i=1}^{m} P(w_i|w_{i-1})$$  \(2\)

and the $P(w_i|w_{i-1})$ is estimated by

$$P(w_i|w_{i-1}) = \frac{c(w_i, w_{i-1})}{c(w_{i-1})}$$  \(3\)

where $c(w_i, w_{i-1}), c(w_{i-1})$ can be computed from a large training corpus.

A. Diacritics restoration problem

In the context of Arabic restoration problem, the diacriticized Arabic word sequence can be assigned a probability score based on a language model. This language model must trained on a large corpus of diacriticized Arabic text. For example, Table 1 shows some possible diacritics alternative for the undiacriticized sentence:

<table>
<thead>
<tr>
<th>علم</th>
<th>الإنسان</th>
<th>ما</th>
<th>لم</th>
<th>يعلم</th>
</tr>
</thead>
<tbody>
<tr>
<td>علم</td>
<td>الإنسان</td>
<td>ما</td>
<td>لم</td>
<td>يعلم</td>
</tr>
<tr>
<td>علم</td>
<td>الإنسان</td>
<td>ما</td>
<td>لم</td>
<td>يعلم</td>
</tr>
<tr>
<td>علم</td>
<td>الإنسان</td>
<td>ما</td>
<td>لم</td>
<td>يعلم</td>
</tr>
<tr>
<td>علم</td>
<td>الإنسان</td>
<td>ما</td>
<td>لم</td>
<td>يعلم</td>
</tr>
<tr>
<td>علم</td>
<td>الإنسان</td>
<td>ما</td>
<td>لم</td>
<td>يعلم</td>
</tr>
<tr>
<td>علم</td>
<td>الإنسان</td>
<td>ما</td>
<td>لم</td>
<td>يعلم</td>
</tr>
<tr>
<td>علم</td>
<td>الإنسان</td>
<td>ما</td>
<td>لم</td>
<td>يعلم</td>
</tr>
</tbody>
</table>

Some of the possible hypotheses for this undiacriticized sentence are:
Based on a language model, a score is assigned to each hypothesis. For example, given a bigram language model, the hypothesis אliwoועקן is assigned a score as follow:

\[ P(H) = P(אliwoועקן)P(אליוועקן)P(אליוועקן)P(אליוועקן)P(אליוועקן) \] (4)

Similarly, all possible hypotheses are scored in a similar way. The best path or the most likely diacritized word sequence is assigned to the hypothesis that has the highest score.

Using this statistical scoring method, Arabic diacritics restoration problem can be defined as a problem of choosing a word sequence \( \hat{H} \) with the maximum language model score given undiacritized word sequence \( W \):

\[ \hat{H} = \arg \max_H P(H|W) \] (5)

A brute force approach to score all hypotheses may be computationally expensive. An efficient algorithm to find the most likely hypothesis is discussed in the following section.

3 DYNAMIC DECODING

An efficient algorithm to find the most likely hypothesis is required to solve the search problem. If the average number of possible alternative diacritics per word is \( N \) and the sequence has \( L \) words, then the complexity of the search problem is related to \( O(N^L) \). Hence, a brute force approach to solve the search problem does not scale.

Fortunately, it is possible to solve the search problem with complexity related to \( O(NL) \) using a dynamic programming algorithm [1]. In this work, we extend the dynamic programming decoding algorithm to support higher order language models. The new search algorithm will depend on dynamic lattices where the scores of different paths will be computed on the run time. Hence, the arcs scores can depend on the decoded history and higher order \( n \)-gram language models can be supported.

The algorithm follows the following steps:

1. A dictionary for undiacritized words (keys) and diacritized word (values) is generated from the training corpus.
2. For an input sequence \( W_1, W_2, \ldots, W_n \), a lattice is created using the dictionary. Each node represents a possible diacritics option. For OOV input word (i.e. a word did not appear in the dictionary), a node is created with the undiacritized word. For example, Figure 1 shows a lattice for the Arabic sentence علم الإنسان ما لم يعلم.
3. The nodes in the adjacent words \( W_n, W_{n-1} \) are connected by arcs, where \( n \) is a discrete time index. The initial nodes of the lattice at \( n = 0 \) are assigned a score computed as a log monogram probability of the diacritized word identifier of the node.
4. At \( n = 1 \), each arc (transition) has a score computed as a log probability of the word identifier of the node given the word identifier of the previous node (bigram probability score).
5. The most likely partial path is computed as in the Viterbi algorithm [11]. Using backtrack, it is possible to the history word sequence for each node.
6. For \( n=2 \) to \( n= \) sentence length, each arc (transition) has a score computed as a log probability of the word identifier of the node given the history word sequence of the previous node. Go to step 5.

The presented algorithm is different from the algorithm described in [1] since the arcs scores are computed at each step using partial history computed as in Viterbi algorithm. We refer to this method as dynamic lattice search.

4 EXPERIMENTS

Automatic diacritics restoration experiments were carried out on the Arabic vocalized text corpus: Tashkeela [12]. The corpus is collected using automatic web crawling methods and it is free. It is collected from Islamic religious heritage books and it contains 6,149,726 words.
In this work, the undiacritized Arabic words and sentences were removed from the corpus. The corpus needs further text normalization steps which are beyond the scope of this work. For example, the word \textit{الإنسان} appears in corpus in two forms due to manual data entry errors. These two words should be normalized to one word. Correcting these mistakes may improve the statistics gathered from the data. The corpus was divided into training and test sets. The properties of the training and test data are summarized in Table 2. The system dictionary was built from the training data and its properties are detailed in Table 3.

The language model was built using the SRILM toolkit [13]. The maximum likelihood (ML) estimation of stochastic language model parameters is based on Equation (3). This estimation is problematic since it assigns zero probabilities to unseen $n$-grams in the training data. Smoothing aims to handle this problem by taking some probability mass from the observed $n$-gram and distribute it to the unseen $n$-grams [14], [15]. Applying smoothing techniques improve dramatically the results [16]. In this work, absolute discounting smoothing method was chosen as it leads to the best results. $D$ parameter for absolute discounting was fixed to $0.5$.

Several model orders were investigated. The model with order 2 contains 794,479 monograms and 10,847,630 bigrams. The correct word count and the diacritization Word Error Rate (WER) are shown in Table 4. The measure WER2 is computed by removing the last diacritic of the words (ignoring the case ending diacritic). Language model with order 4 leads to the best results. In addition, WER2 measure shows that most diacritization errors are related to the syntax of Arabic language.

5 CONCLUSIONS

In dynamic programming Arabic diacritics retoration, the possible word sequences with diacritics are assigned scores using statistical $n$-gram language modeling approach. Using the assigned scores, it is possible to search the most likely sequence using a dynamic programming algorithm.

In this paper, we develop a new search algorithm that supports higher order $n$-gram language models. The new search algorithm will depend on dynamic lattices where the scores of different paths will be computed on the run time. Hence, the arcs scores can depend on the decoded history and higher order $n$-gram language models can be supported. Preliminary results on a public domain corpus show that dynamic programming decoding based on higher order $n$-gram
models can lead to better results than bigram models.

References


