Omni Font OCR Error Correction with Effect on Retrieval

Walid Magdy*  
School of Computing  
Dublin City University  
Dublin, Ireland  
wmagdy@computing.dcu.ie

Kareem Darwish  
Cairo Microsoft Innovation Center  
Microsoft Research  
Cairo, Egypt  
kareemdm@microsoft.com

Abstract—Recent library digitization projects attempt to provide large collections of printed material from varying sources in a searchable format. The scanned documents are typically processed using Optical Character Recognition (OCR), which typically introduces errors in the text. This paper proposes a technique for correction of OCR degraded text that is independent of character-level OCR errors, and hence independent of scanned document source. It is based on language modeling in conjunction with a uniform character error model. The correction method relies on a uniformly distributed character error model based on the edit distance between a misrecognized word and a candidate correction with the assistance of a domain specific language model. This approach is well suited for situations where document are obtained from a variety of sources. The approach is compared to previously reported approaches that use character level models to test correction effectiveness and consequent retrieval effectiveness. Although the approach is tested on Arabic OCR text documents, the approach is potentially applicable to text that is degraded using different processes from different languages.

The paper will be organized as follows: Section 2 provides background information on Arabic OCR and OCR error correction; Section 3 describes the data set; Section 4 presents the error correction methodology; Section 5 reports and discusses experimental results; and Section 6 concludes the paper and provides possible future directions.

I. INTRODUCTION

Recent advances in printed document digitization and processing led to large scale digitization efforts of legacy printed documents producing document images. To enable subsequent processing and retrieval, the document images are often transformed to character-coded text using Optical Character Recognition (OCR). Although OCR is fast, OCR output typically contains errors. The errors are even more pronounced in OCR'ed Arabic text due to Arabic's orthographic and morphological properties. The introduced errors adversely affect linguistic processing and retrieval of OCR'ed documents. Recent work in OCR correction depends on the presence of a source specific character error model for the OCR output text, which makes the correction systems depend on font, OCR system, and printed document quality and requires training examples to build error models [21, 22]. In applications where documents are obtained from heterogeneous sources, building a character level model for every font type and size and every degradation level is impractical. This paper introduces a general method for error correction that does not require the training of a character error model. The correction method relies on a uniformly distributed character error model based on the edit distance between a misrecognized word and a candidate correction with the assistance of a domain specific language model. This approach is well suited for situations where document are obtained from a variety of sources. The approach is compared to previously reported approaches that use character level models to test correction effectiveness and consequent retrieval effectiveness. Although the approach is tested on Arabic OCR text documents, the approach is potentially applicable to text that is degraded using different processes from different languages.

II. BACKGROUND

OCR transforms a document image into character-coded text, typically using automatic classification. The character error rate can be influenced by the scanned copy quality, the scanning resolution, and any mismatch between recognized text and the classifier’s training set. Arabic OCR presents several challenges, including Arabic’s cursive script, the optional use of word elongations, ligatures, and diacritics, the presence of dots in 15 of the 28 letters in the Arabic alphabet[10], and Arabic’s morphological complexity [4]. There are a number of commercial Arabic OCR systems, with Sakhr’s Automatic Reader and Shonut’s Omni Page being the most widely used. Retrieval of OCR degraded text documents has been reported for many languages, including English [16], Chinese [31], and Arabic [10].

Much research has been done to correct recognition errors in OCR-degraded collections. There are two main approaches to error correction, namely word-level and passage-level. Some of the kinds of word level post-processing include the use of dictionary lookup [8, 9, 17, 18], character [20, 29] and word n-gram [17, 21] frequency analysis, and morphological analysis [13, 27]. Passage-level post-processing techniques include the use of word n-grams [21], word collocations [17], grammar [3] (which is challenging due to current poorness of Arabic parsing [25]), conceptual closeness [17], passage level word clustering [29]...
(which requires handling of affixes for Arabic [12]), and linguistic and visual context [17]. Dictionary lookup, which is the basis for this paper, is used to compare recognized words with words in a lexicon [8, 9, 17, 18]. Finding the closest matches to every OCR’ed word in the dictionary is attempted, and the matches are then ordered using a character level error model in conjunction with either a unigram probability of the matches in text [18] or a n-gram language model [21].

Concerning Arabic, most early studies of character-coded Arabic text retrieval relied on relatively small test collections [2, 6], which suggested that using roots as index terms was best. However, more recent results that are based on a single large collection (from TREC-2001/2002) [15, 26] suggested that using lightly stemmed words or character 3 and 4-grams was best. The effects of normalizing alternative characters, removal of diacritics and stop-word removal have also been explored [11, 14, 19, 23, 24]. As for OCR degraded Arabic text, a previous study suggests that 3 and 4-character grams and their combinations with index terms obtained through morphological analysis, such light stems, outperform all other kinds of index terms [11]. Another recent study suggested that modest Arabic OCR correction had insignificant impact on Arabic retrieval effectiveness [22].

III. DATA SET

To test the correction system, two document collections from two different domains were used. The first collection consists of 2,730 documents, which were obtained from a modern printing of a medieval religious Arabic book (called “The Provisions of the Return” or “Provisions” for short by Ibn Al-Qayim) [21]; for brevity, it is referred to here simply as the ZAD collection. One of the advantages of the ZAD collection is the presence of an error-free (clean) character-coded copy of it. The Provisions book was scanned at 300x300 dots per inch (dpi), and Sakhr’s Automatic Reader was used to OCR the scanned pages. Associated with the collection are 25 topics and relevance judgments, which were built by exhaustively searching the collection. The number of relevant documents per topic ranges from 3 to 72, averaging 20. The average query length is 5.4 words [10]. To compare the proposed approach to an approach that uses a trained source-specific character level model, 4,000 words were randomly picked from the collection and were then manually corrected, and the corrupted and manually corrected versions were aligned. The Word Error Rate (WER) for the 4,000 testing words was 39%. Another set of sentences that are collectively made-up of 2,000 words were picked, corrected, and set aside for testing.

The second collection is the Text Retrieval Conference (TREC) 2002 Cross-Language IR (CLIR) track collection; for brevity, it is referred to here simply as the TREC collection. It contains 383,872 articles from the Agence France Press (AFP) Arabic newswire. Associated with the collection are 50 topics with appropriate relevance judgments. A unigram model was used to degrade the collection in a way that simulates OCR degradation. OCR degradation was modeled as a noisy channel in which the observed characters result from the application of some distortion function on the real characters. The model used here accounts for three character edit operations: insertion, deletion, and substitution. Formally, given a clean word #C1..Ci..Cn# and the resulting word after OCR degradation #D1..Dj..Dm#, where Dj resulted from Ci, ε representing the null character, L representing the position of the letter in the word (beginning, middle, end, or isolated), and # marking word boundaries, the probability estimates for the three edit operations for the models, are:

\[
P_{\text{substitution}} \left( C_i \rightarrow D_j \right) = \frac{\text{count} \left( C_i \rightarrow D_j | L_{C_i} \right)}{\text{count} \left( C_i | L_{C_i} \right)} \quad (1a)
\]

\[
P_{\text{deletion}} \left( C_i \rightarrow \epsilon \right) = \frac{\text{count} \left( C_i \rightarrow \epsilon | L_{C_i} \right)}{\text{count} \left( C_i | L_{C_i} \right)} \quad (1b)
\]

\[
P_{\text{insertion}} \left( \epsilon \rightarrow D_j \right) = \frac{\text{count} \left( \epsilon \rightarrow D_j \right)}{\text{count} \left( \epsilon \right)} \quad (1c)
\]

The model was trained using the 2,000 aligned training words from the ZAD collection. The resulting garbler reads in a clean word #C1..Ci..Cn# and synthesizes OCR degradation to produce #D1..Dj..Dm#. For a given character Cn, the garbler chooses a single edit operation to perform by sampling the estimated probability distribution over the possible edit operations. If an insertion operation is chosen, the model picks a character to be inserted prior to Cj by sampling the estimated probability distribution for possible insertions. Insertions before the # (end-of-word) marker are also allowed. If a substitution operation is chosen, the substituted character is selected by sampling the probability distribution of possible substitutions. The deletion operation simply delete the chosen character. After degradation, a set of 4,000 words were randomly picked and were found to have a 31% WER. The words were manually corrected to train a character level model to compare correction with and without training such a model. Another set of sentences, composed collectively of 6,000 words, was randomly picked, corrected, and set aside for testing. For all words in both collections, the different forms of alef (hamza, alef, alef maad, alef with hamza on top, hamza on wa, alef with hamza on the bottom, and hamza on ya) were normalized to alef, and ya and alef maqsoura were normalized to ya.

IV. ERROR CORRECTION METHODOLOGY

Since the proposed approach does not use a trained source-specific character-level error model, Levenshtein edit
distance is used instead with uniform probability distribution for different edit operations. In other words, all substitutions, deletions, and insertions are considered equally likely. For a given OCR’ed word \( w_{OCR} \), a dictionary is checked and the closest candidate correction \( W_{\text{cand}} = \{ w_j , \ldots , w_i , \ldots w_m \} \) are ranked according to their edit distance and unigram probability of observing a word in text according to the following similarity function \( S_{ED} \):

\[
S_{ED}(w_i) = e^{-CE(ED(w_{OCR}, w_i))} \cdot \frac{P(w_i)}{P(w_i)} \quad (2)
\]

where \( ED \) is edit distance between \( w_{OCR} \) and \( w_i \), \( P(w_i) \) is the unigram probability of \( w_i \) in the dictionary, and \( C \) is a scaling factor to affect the relative contribution of edit distance (\( C \) is proportional to the effect of edit distance). This will be referred to as the “ED” model.

Best \( N \) candidates will be selected according to the previous formula and a language model is used to select the best candidate correction according to context.

To properly compare to state-of-the-art correction, an alternative segment based character error model was trained as described by Magdy and Darwish [21]. This model will henceforth be referred to as the “REF” model. Formally, for a given degraded word \( w_{OCR} = \#D_1..D_m\), \( D_i..D_j\# \), a set of possible correction \( W_{\text{cand}} = \{ w_i , \ldots , w_i , \ldots w_m \} \), where \( w_i = #C_{i\#}..C_{j\#}..C_{m\#} \), the null character \( \# \), and the word boundary marker \( \# \), the probability estimates for the three edit operations for the models are:

\[
P_{\text{substitution}}(C_k..C_l \rightarrow D_i..D_j) = \frac{\text{count}(C_k..C_l \rightarrow D_i..D_j)}{\text{count}(C_k..C_l)} \quad (3a)
\]

\[
P_{\text{deletion}}(C_k..C_l \rightarrow \#) = \frac{\text{count}(C_k..C_l \rightarrow \#)}{\text{count}(C_k..C_l)} \quad (3b)
\]

\[
P_{\text{insertion}}(\# \rightarrow D_i..D_j) = \frac{\text{count}(\# \rightarrow D_i..D_j)}{\text{count}(\#)} \quad (3c)
\]

The similarity function \( S_{REF} \) between the \( w_{OCR} \) and a candidate correction \( w_i \) combines the character transformation probability with the unigram probability of observing the proposed correction in the text as follows:

\[
S_{REF}(w_i) = \prod_{\text{all}D_i..D_j} P(D_i..D_j | C_k..C_l) \cdot \frac{P(w_i)}{P(w_i)} \quad (4)
\]

A. Candidates Selection

To get \( m \) initial candidates that are similar to the OCR’ed word, all the words in dictionary are indexed as combinations between letters unigrams, bigrams, trigrams, and the word length. For example: “example” \( \rightarrow \{ e , x , a , m , p , l , e , #e , ex , xa , am , mp , pl , le , e#, #ex , exa , xam , amp , mpl , ple , le#, <NO>7</NO> \} \). A given OCR’ed word is used as a query with the same format, but instead the word length will be a range from length-1 to length+1 to allow the presence of deletion or insertion of characters. For experiments in this paper, the Indri search toolkit [1] was used to index the dictionary and run queries. For each OCR word, the top 1,000 (\( m = 1,000 \)) retrieved words are scored according to the similarity function.

B. Constant “C” selection

In order to obtain the best value of the scaling factor \( C \) in equation (2), some offline correction experiments were performed with different values of \( C \), namely 1 to 8. Experiments were performed on the ZAD collection, and the presence of the proper correction among best \( N \) candidate corrections were noticed as shown in Figure 1.

From graph, it is clear that the probability to find the proper correction increases as more candidates are taken, and the accuracy nearly saturates after \( N = 10 \). As shown in the graph, higher values for \( C \) give better performance than lower values, and the best performance was at \( C = 5 \) where the accuracy reached 83.8\% at \( N = 5 \), 86\% at \( N = 10 \), and 87.3\% at \( N = 20 \). For the remainder for the paper, \( C \) will be used with a value of 5.

![Fig. 1. Accuracy vs. number of candidate corrections for ZAD set with different values of C. Accuracy refers to the presence of a proper correction among the \( N \) best candidate corrections](image)

C. Language Modeling

For language modeling, a trigram language model was trained without any kind of morphological processing. The language model was built using the SRILM toolkit [5] with Good-Turing smoothing and default backoff.

Given a corrupted word sequence \( \Delta = \{ \delta_1 , \ldots , \delta_i , \ldots , \delta_m \} \) and \( X = \{ X_1 , \ldots , X_i , \ldots , X_m \} \), \( X_i = \{ \chi_{i0} , \ldots , \chi_{iN} \} \) are possible candidate corrections of \( \delta_i \) (\( N \) is the number of candidates corrections taken), the aim was to find a sequence \( \Omega = \{ \omega_i , \ldots , \omega_k , \ldots , \omega_m \} \), where \( \omega_i \in X_i \), that maximizes:
where $\beta$ is the scaling factor to affect the relative contribution of the edit distance ($\beta$ is proportional to the effect of edit distance).

When combining language modeling with the REF model, the goal is to maximize [21]:

$$
\left( \prod_{j=1 \ldots m, j=1 \ldots N} P(\chi^*_{j, i} | \chi_{i-1, j}, \chi_{i-2, j}) \right) \prod_{i \in D, j \in D} P(D_i | C_i, C) \quad (6)
$$

D. Testing Correction Effectiveness

To test the effectiveness of correction, two types of tests were performed. The first examined the reduction in word error rate, and the second observed the effect of correction on retrieval effectiveness. The first test was applied to both collections, while the later was applied to the ZAD collection only. The reasons why the second test was not performed on the TREC collection are explained later.

In examining the reduction in word error rate for the ZAD and TREC collections, the top $N$ candidate corrections, with $N$ varying between 1 and 20, are examined to determine if the proper correction is among them. When using language modeling, the effect of the scaling factor $\beta$, which is proportional to the effect of edit distance, is examined at different values of $\beta (\beta = \{1, 2, 4, 8\})$ and the top correction being considered by the language model were either 5 or 10.

For building a language model for the ZAD collection, a web-mined collection containing most of the books of Ibn Taymia (the teacher of the author of Provisions book) were used to train a tri-gram language model. For the TREC collection, all the text from the TREC collection that was not part of the character level model training set and not from the test set was used to build a language model, which will be referred to henceforth as the AFP-LM model. Another language model was trained from a web-mined collection of Arabic newswire articles from the BBC, Al-Ahram newspaper, Al-Jazeera news website, Al-Wafd newspaper, and Al-Moheet news website. This language model will be referred to as the News-LM model. Unfortunately, the news articles in this collection do not span the same time period as the TREC collection.

As mentioned earlier, correction effectiveness was tested on sets of 2,000 and 6,000 words for the ZAD and TREC collections respectively.

The effect of correction on retrieval effectiveness was examined for the ZAD collection. The retrieval experiments were performed on the clean, corrupted, and corrected versions of the ZAD collection described above. The versions of the collection were indexed and searched using words, character 3-grams, character 4-grams, and lightly stemmed words obtained using Al-Stem [15]. For all experiments, Indri was used with default parameters with no blind relevance feedback. The figure of merit for evaluating retrieval results was mean average precision (MAP). Statistical significance between different retrieval results was performed using a paired 2-tailed t-test and a Wilcoxon test with continuity correction with $p$-values $\leq 0.05$ to assume statistical significance. The Wilcoxon test $p$-values are being reported for completeness. There are some indications that the t-test is sufficiently reliable despite the fact that the normality condition might not be met [28].

V. EXPERIMENTAL RESULTS

Table 1 reports the percentage of words for which a proper correction was not found in the top $N$ generated corrections for both test sets using the ED and REF models. The change in percentage of words for which correction failed is faster with increasing $N$ for the ED model compared to the REF model. This results in a narrowing of the gap between the two models for the ZAD collection from 6.2% difference to 2.2% difference. The difference between the percentages for varying values of $N$ for the TREC collection was surprising small with the ED model slightly outperforming the REF model for large $N$'s. These results are promising, because they suggest that using a language model to aid in picking the most likely correction is likely to lessen the impact of not using a trained character model. Further, the chances of finding proper corrections beyond 10 corrections are greatly diminished.

Tables 2 shows the effect of using a trigram language model in conjunction with edit distance in reranking the top 5 and top 10 candidate corrections with different values of $\beta$ for the ZAD and TREC test set respectively. The results show that using the top 10 corrections is better than using just the top 5. The best values for $\beta$ were 2 and 4 for the ZAD and TREC sets respectively. For TREC set, results highlight the fact that the use of a better language model, such as the AFP LM that is trained on a set that matches style and temporal coverage of the text to be corrected, yields better correction effectiveness compared to the use of another less matching language model such the News LM model. In fact, Table 3 shows that compared to using no language modeling, utilizing the AFP LM had a visible effect on WER (more than 5% drop in WER), while utilizing the News LM had minimal effect on WER (less than 1% drop in WER). Since using the AFP LM for correcting the whole AFP collection would not be appropriate (it would tantamount to using the same set for training and testing) and the use of the News LM yields minimal improvements, the IR experiments were done for the ZAD collection only.
Table I
PERCENTAGE OF WORDS FOR WHICH PROPER CORRECTION WAS NOT FOUND IN TOP N CORRECTIONS

<table>
<thead>
<tr>
<th># of corrections</th>
<th>Model 1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZAD WER = 39%</td>
<td>ED</td>
<td>28.4</td>
<td>21.4</td>
<td>18.8</td>
<td>16.2</td>
<td>14.0</td>
</tr>
<tr>
<td></td>
<td>REF</td>
<td>22.2</td>
<td>16.9</td>
<td>15.0</td>
<td>13.2</td>
<td>11.5</td>
</tr>
<tr>
<td>TREC WER = 31%</td>
<td>ED</td>
<td>12.6</td>
<td>7.6</td>
<td>5.7</td>
<td>4.2</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>REF</td>
<td>11.1</td>
<td>6.6</td>
<td>5.6</td>
<td>4.6</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Table II
WER FOR DIFFERENT VALUES OF β FOR ZAD AND TREC TEST SETS

<table>
<thead>
<tr>
<th></th>
<th>ZAD</th>
<th>TREC</th>
</tr>
</thead>
<tbody>
<tr>
<td>β</td>
<td>WER</td>
<td>WER</td>
</tr>
<tr>
<td></td>
<td>(AFP LM)</td>
<td>(News LM)</td>
</tr>
<tr>
<td>5</td>
<td>23.8%</td>
<td>12.8%</td>
</tr>
<tr>
<td>10</td>
<td>21.0%</td>
<td>13.2%</td>
</tr>
<tr>
<td>1</td>
<td>20.2%</td>
<td>15.6%</td>
</tr>
<tr>
<td>2</td>
<td>17.2%</td>
<td>16.2%</td>
</tr>
<tr>
<td>4</td>
<td>21.3%</td>
<td>11.8%</td>
</tr>
<tr>
<td>8</td>
<td>34.5%</td>
<td>15.0%</td>
</tr>
</tbody>
</table>

Table 3 compares the correction effectiveness when employing a trained vs. a uniform character error model with and without LM for the ZAD and TREC collections respectively. The results show that using a trained character level model yields noticeably better correction effectiveness compared to using the proposed uniform character error model. However, as will be shown later, the difference in effect on retrieval effectiveness is less dramatic.

For IR experiments, language modeling was used to correct the ZAD collection with N=10, and the corrected versions (with ED and REF models) were compared to each other and to the clean and original OCR’ed versions. For the reasons mentioned earlier, IR experiments were performed on the ZAD collection only. Figure 2 summarizes the retrieval results of searching the clean, OCR’ed (bad), and corrected (with character model and edit distance) versions of the ZAD collection using words, light stems, character 3-grams, and character 4-grams. Comparing the retrieval effectiveness when using language modeling with the ED and REF models has showed to be both statistically indistinguishable for different index terms. Table 4 provides the p-values of the paired 2-tailed t-tests and Wilcoxon tests of comparing the results for both corrections models with language modeling to the clean and original OCR’ed versions. The results confirm that character 3- and 4-grams are indeed the best index terms with 3-grams on uncorrected text outperforming words and light stems even after correction. For both correction models, character 3-grams – as an index term – achieved the highest MAP and error correction statistically significantly improved retrieval effectiveness, and retrieval effectiveness was statistically indistinguishable from the effectiveness of retrieving from the clean version. The same was true for character 4-grams when using the REF model with language modeling. Contrary to the reports in the literature [22], the results suggest that “good” error correction with and without a source-specific character model could statistically significantly improve retrieval effectiveness, and possibly be statistically indistinguishable from retrieving clean version.

Table III
COMPARING CORRECTION EFFECTIVENESS WITH AND WITHOUT USING CHARACTER MODEL FOR ZAD AND TREC SETS

<table>
<thead>
<tr>
<th></th>
<th>Uniform Character Model</th>
<th>WER</th>
<th>Error Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZAD Set</td>
<td>Baseline</td>
<td>28%</td>
<td>27%</td>
</tr>
<tr>
<td></td>
<td>LM</td>
<td>17%</td>
<td>56%</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td>22%</td>
<td>44%</td>
</tr>
<tr>
<td></td>
<td>LM</td>
<td>12%</td>
<td>70%</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td>12.6%</td>
<td>59%</td>
</tr>
<tr>
<td></td>
<td>AFP LM</td>
<td>7.3%</td>
<td>76%</td>
</tr>
<tr>
<td></td>
<td>News LM</td>
<td>11.7%</td>
<td>62%</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td>11.1%</td>
<td>64%</td>
</tr>
<tr>
<td></td>
<td>AFP LM</td>
<td>5.9%</td>
<td>81%</td>
</tr>
<tr>
<td></td>
<td>News LM</td>
<td>10.7%</td>
<td>65%</td>
</tr>
</tbody>
</table>

The results suggest that training a character error model yields more effective correction, but the effect of correction on retrieval effectiveness is uncertain. Further, training a character error model is often disadvantageous due to its dependency on font size and type, OCR system, scanned paper quality, and other factors. On the other hand, the correction technique proposed in this paper does not require the training of character level models and achieves comparable retrieval results. This is approach is more practical for applications where printed pages are obtained from a variety of heterogeneous sources.
VI. CONCLUSION AND FUTURE WORK

The paper examined a technique for OCR error correction based on language modeling and a uniform character model that uses edit distance only and compares to state-of-the-art correction techniques based on language modeling and trained character error level models. Although the proposed technique yielded lower correction effectiveness, its impact on retrieval effectiveness is statistically significant and at par with state-of-the-art correction techniques. The main requirement of the proposed technique is the training of a “good” language model matching genre, style, and temporal coverage. The advantage of using a character model independent technique is clear in applications were printed documents vary in source, font, and degradation level and are potentially scanned and OCR’ed using different systems.

For future work, the proposed technique needs to be tested on heterogeneous printed sources and potentially other degradation sources such as automatic speech recognition. A Factored language model [32] might prove beneficial to degradation sources such as automatic speech recognition. This can be instrumental in overcoming the out-of-vocabulary problem.

REFERENCES


