International Journal on Document Analysis and Recognition (IJDAR)
https://doi.org/10.1007/s10032-018-0297-y

ORIGINAL PAPER

A novel Arabic OCR post-processing using rule-based and word context techniques

Iyad Abu Doush[1,2] · Faisal Alkhateeb[2] · Anwaar Hamdi Gharaibeh[2]

Received: 6 February 2017 / Revised: 14 February 2018 / Accepted: 22 March 2018

Abstract
Optical character recognition (OCR) is the process of recognizing characters automatically from scanned documents for editing, indexing, searching, and reducing the storage space. The resulted text from the OCR usually does not match the text in the original document. In order to minimize the number of incorrect words in the obtained text, OCR post-processing approaches can be used. Correcting OCR errors is more complicated when we are dealing with the Arabic language because of its complexity such as connected letters, different letters may have the same shape, and the same letter may have different forms. This paper provides a statistical Arabic language model and post-processing techniques based on hybridizing the error model approach with the context approach. The proposed model is language independent and non-constrained with the string length. To the best of our knowledge, this is the first end-to-end OCR post-processing model that is applied to the Arabic language. In order to train the proposed model, we build Arabic OCR context database which contains 9000 images of Arabic text. Also, the evaluation of the OCR post-processing system results is automated using our novel alignment technique which is called fast automatic hashing text alignment. Our experimental results show that the rule-based system improves the word error rate from 24.02% to become 20.26% by using a training data set of 1000 images. On the other hand, after this training, we apply the rule-based system on 500 images as a testing dataset and the word error rate is improved from 14.95% to become 14.53%. The proposed hybrid OCR post-processing system improves the results based on using 1000 training images from a word error rate of 24.02% to become 18.96%. After training the hybrid system, we used 500 images for testing and the results show that the word error rate enhanced from 14.95 to become 14.42. The obtained results show that the proposed hybrid system outperforms the rule-based system.

Keywords Automatic post-processing · Arabic OCR post-processing · Language model · Alignment technique · Error model

1 Introduction

The number of scanned documents in the Arabic language is huge. Also, many books are available in a non-text format (e.g., as a PDF or as images), which could not be converted easily into text. The ability to convert such documents into text will make the information searchable and will enable the conversion of the document into another format (e.g., audio).

Optical character recognition (OCR) can be used to convert an image which contains text into an editable text [7]. A large number of documents (e.g., books) in non-text format and the satisfactory OCR accuracy for the digitalized documents increase the number of converted images into text. Numerous book digitalization projects have used OCR systems, for example, Improving Access to Text (IMPACT) project, the TextGrid project, and the semiautomatic Arabic DAISY books project [4].

Printed books may contain many challenges when they are converted into text such as different fonts, colors, and touching character at the edges. This makes the text segmentation not an easy task. In addition, after scanning the document the obtained images may contain blur and/or curved lines. These challenges may produce a text document with many errors [45]. In order to minimize these errors, the OCR post-processing methods are used. The main goals of

✉ Iyad Abu Doush
iyad.doush@yu.edu.jo

Faisal Alkhateeb
alkhateebf@yu.edu.jo

[1] Computer Science and Information Systems Department, American University of Kuwait, Salmiya, Kuwait

[2] Computer Sciences Department, Yarmouk University, Irbid 21163, Jordan

Published online: 05 April 2018

applying the OCR post-processing methods are error detection and correction. Different approaches are used for the OCR post-processing which are manual error correction, dictionary-based error correction, and context-based error correction [15,28].

In this paper, we use the context-based error correction approach as it provides better results when compared with the other techniques. A small amount of research work has been done on developing OCR post-processing techniques for the Arabic language. The Arabic OCR results show high error rates due to several factors [2,6,28].

The OCR alignment technique is the process that aims to match the ground truth text with the OCR misrecognized text. Manual and automatic approaches could be used for the OCR alignment. Manual alignment relies on a group of people to match the ground truth and the misrecognized text [6]. Many researchers suggested automatic alignment techniques for different languages [14,43,46,47]. However, the use of automatic alignment approaches does not guarantee the full match between the ground truth and the misrecognized text.

The OCR rule-based system [12] uses weighted finite-state transducers (WFST) to correct OCR errors. The OCR result is compared with the ground truth to select the appropriate edit operation (i.e., insertions, deletions, and substitutions). The developed set of operations is then used to form the rules to correct the misrecognized text.

This research, which is based on the thesis work [21], applies language model and error model techniques along with Google's suggestion to improve Arabic OCR post-processing by proposing a novel post-processing system for Arabic documents. The proposed system is compared with the rule-based system introduced by [12]. Moreover, we propose a new fast and accurate text alignment technique called FAHTA to automatically evaluate the results against the ground truth, and the OCR system or the post-processing system. In this paper, we create the first large-sized open Arabic OCR image database with its ground truth.

The remainder of this paper is organized as follows; Sect. 2 introduces the research background. Section 3 is dedicated to the presentation of the proposed methodology. Experimental results are discussed in Sect. 4. The concluding remarks, as well as the future work, are presented in Sect. 5.

2 Background

2.1 Characteristics of Arabic letters

The Arabic language is the national language in 26 countries. It is ranked fourth, based on the number of internet users in 2013. The Arabic language has 29 letters; each letter has different characteristics [35]. The main features of the Arabic printed letters according to [1,5,9,26,27] are:

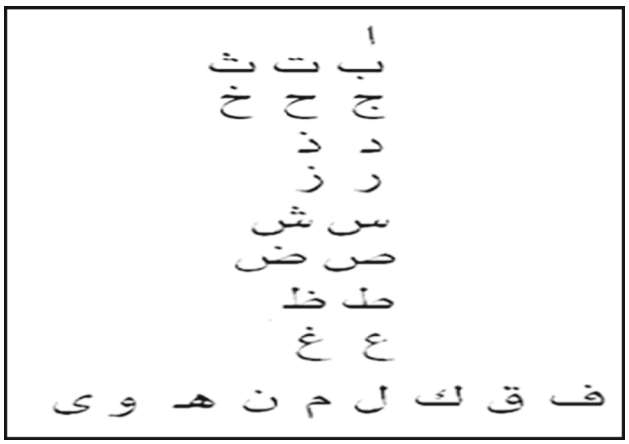

Fig. 1 Arabic alphabet [6]

- Arabic words are written right to left.
- A letter may differ in its shape depending on its position in the word. A letter may have four possible shapes: isolated, initial, middle, and final.
- Some letters have the same shape, but they differ in the place of the dots (e.g., the letters (ج,خ) je, kha)), the number of the dots (e.g., the letters (ت, ث) (ta, tha)), or both (e.g., the letters (ب, ت, ث) (ba, ta, tha)). Figure 1 shows the set of Arabic letters.
- Other characters exist in Arabic such as Ta-Marbuta and Alif-Maqsurah. Ta-Marbuta has two shapes: isolated (ة) (ta) and final (ـة) (ta). On the other hand, Alif-Maqsurah has the two shapes: isolated (ى) (a) and final (ـى) (a).
- The Arabic script may contain a vowel or diacritic mark. Each diacritic mark could be under or above the Arabic character.

2.2 OCR post-processing

The OCR system has the following five steps: pre-processing, character segmentation, classification, recognition, and post-processing. The output of each stage is the input of the next stage (see Fig. 2). The overall performance of the OCR system is driven by each one of these stages [6,8,37].

OCR is a process that converts an image of handwritten or printed text into a digital text. This obtained text may have different types of errors. The main types of OCR word errors are non-word errors and real-word errors. A non-word error is an invalid language word, while a real-word error is a valid language word but not the word from the original text [20,28]. In order to eliminate or reduce these errors, OCR post-processing techniques can be used.

Different techniques are used for OCR post-processing including manual error correction, dictionary-based error

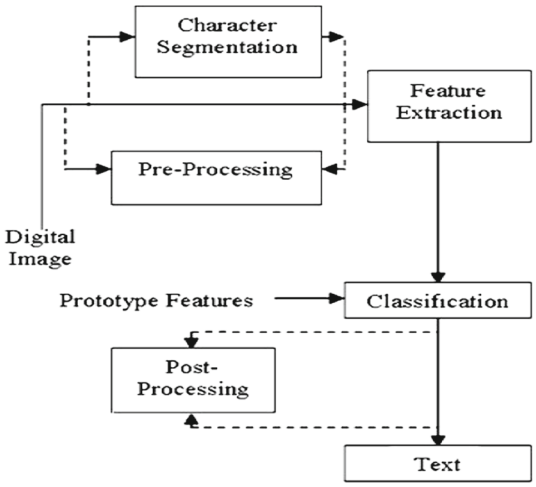

Fig. 2 A typical OCR system [6]

correction, and context-based error correction [15,28]. Manual error correction or proof-reading is the simplest way which selects a group of people to correct the OCR output manually. This technique can solve non-word errors and real-word errors. This technique is not only time consuming, but also it cannot be applied for a large input data. Dictionary-based error correction approach can handle the non-word errors by matching the error word with the suggested words in a lexicon. Specific techniques can be used to search for the lexicon efficiently [24]. However, string search methods are not efficient when we have a large dictionary. Therefore, n-grams and weighted finite-state machines techniques may be used instead.

The context-based error correction approach is integrated with a dictionary-based error correction approach by taking the context of the text into consideration when correcting the errors. It aims to solve real-word errors based on the language grammar and based on the word context. However, many researchers use dictionary-based error correction and context-based error correction to handle the two types of OCR word errors such as [16,19,34]. Statistical Language Model which consists of finite word forms by using a big corpus is used by other researchers [12,23,32].

2.3 Language and error model

Language model (LM) is a probability distribution for a sequence of words in a document [25]. It is used in many natural language processing applications such as automatic speech recognition (ASR), machine translation (MT), OCR, and handwritten recognition. However, the goal of the LM is to define the probability of words in a document to predict the likelihood of observing any query. In addition, the ranking in the LM is satisfied by ordering these probabilities [31].

The main goal of using OCR post-processing is to minimize the OCR error rate. To achieve this goal, LM can be used as it can suggest the correct words in an efficient manner. However, with a big size dictionary, the traditional string search techniques cannot work efficiently. In addition, for unknown word form, they cannot suggest the predicted word form. In order to overcome these drawbacks, error model can be used.

The error model uses the Levenshtein edit distance [30]. The goal of edit distance is to find the similarity between two strings by using a finite number of the three operations: insertion, deletion, and modification. Edit distance value is used to control the search in a big size dictionary [40]. Schulz and Mihhov proposed the enhancement of the traditional edit distance by using a two-sided finite-state automaton.

In another work, Al Azawi and Breuel [12] proposed an error model using the weighted two-sided finite-state automata transducers for OCR post-processing. The proposed model depends on the context of the confusions rules which are extracted by identifying the operations needed to correct the string when the OCR result is compared with the ground truth. The error model proposed by AL Azawi and Breuel is used in this study.

2.4 Arabic OCR database

The OCR is used to transform a big-sized image into a small-sized text document. In order to provide an accurate conversion result for the OCR, we need to use a database for the training phase. The OCR database type is based on the OCR stage. Hence, the edited database and the image database are the main OCR database types.

Many research works are available on the edited word Arabic database. Najoua and Noureddine [36] created 1000 printed words with different font sizes and types. In another work, 100 multi-font words were tested by [10]. Broumandnia et al. [18] created a database with 1000 words with different sizes and fonts. Lee et al. [29] proposed a big Arabic Language Model which contains 110,000 segmented words from a 15 million unsegmented words Arabic corpus. Furthermore, Habeeb et al. [23] proposed a statistical language model which is extracted from Wikipedia's database in the form of an XML file.

Slimane et al. [41,42] published the standard Arabic Printed Text Images (APTI) which is an example of the word image database type. This database contains 45 million single-word images. Each word image has its own ground truth in the form of the XML file.

Schlosser [39] prepared 750-page images from different books and magazines. However, this database is not available for the public.

AbdelRaouf et al. [2] proposed a multimodal Arabic corpus (MMAC) that can be used in both OCR development and

linguistics. The total number of words is 282,593. The Arabic corpus was built from Arabic Web sites, Arabic news Web sites and Arabic–Arabic dictionaries. Recently, a dataset for Arabic OCR with 1,833 images is released with the name 'BCE-Arabic-v1' [38]. However, its size is not enough to build the language model and error model.

To the best of our knowledge, there is no Arabic OCR database that is available to be used by the research community with its ground truth with a size that is suitable to build a robust language model and error model. Hence, our work provides an Arabic document OCR database that simulates UW-III database [22]. UW-III is the third edition of the English and Japanese Document OCR. The ground truth was encoded with ASCII and the size is up to 9000 images.

2.5 OCR post-processing systems

Several OCR post-processing systems are proposed in the literature. Magdy and Darwish [33] built an Arabic OCR error correction system by using three techniques OCR character level models, a language model with 10 maximum possible corrections and using shallow morphology.

Bassil and Alwani [15] proposed an error correction algorithm which aims to detect and correct non-word error and real-word error for low-quality image document. It is based on Google's online Spell Checker. This study is evaluated using English and Arabic documents. The algorithm reduced the error rate for the English language from 21.4 to 3.1%. On the other hand, the error rate is reduced for the Arabic language from 12.5 to 3.1%.

Habeeb et al. [23] proposed Arabic OCR post-processing method based on the 2-g language model. It consists of two parts: the extraction part, and the correction part. This algorithm reduced the error rate from 7.81 to 2.29%, while it reduced the error rate from 7.81 to 6.09% for non-word error.

Al Azawi and Breuel [12] proposed techniques to correct the OCR errors based on a weighted finite-state transducer (WFST) with context-dependent rules. In their experiment, they tested the technique with the English language and compared the results with the single-character rule-based approach. The error rate is 0.68%, while the baseline is 1.14% and the error rate of the single-character rule-based approach is 1.0%. Al Azawi et al. [13] proposed another language-independent technique to improve the accuracy of the technique. They built a language model using recurrent neural network (RNN) technique. The experimental results show that this technique enhanced the error rate from 0.68 to 0.46% for the English language, and from 3.8 to 1.58% for the Urdu language.

Abu Doush and Al-trad [3] proposed an Arabic OCR post-processing system based on three different scenarios: Google's online spelling system, Microsoft Office Word with Google's online spelling system and Ayaspell spell checker with Google's online spelling system. The accuracy rates were 37, 49, and 28% respectively, for each scenario.

2.6 Discussion

The post-processing is considered in the error correction stage of the OCR system. Some techniques are based on proof-reading, dictionary, and context. The problem of Arabic character error correction is mainly due to the huge number of words which have a similar shape. This happened because several Arabic letters share the same shape. Dictionary methods take a long time to find the optimal solution. In addition, it corrects only non-word errors. The context approach applies the training phase on the context dataset. It requires a big-sized dataset to learn. Unfortunately, no large Arabic OCR context database is available to be used by the research community. Hence, the Arabic OCR post-processing field needs a big Arabic database which is developed for our study and it can be used by other researchers in the future.

Some researchers use WFSTs to reduce the computation time. For instance, if the word has separated or connected letters then this problem could be solved by using context rules extracted from the error model [12,13]. In addition, the n-gram language model (LM) is used to suggest the correction of the words which can be defined as follows [31]:

Let $\mathcal{S}$ be a sequence composed of r consecutive index terms ($\mathcal{K}_i$). That is,

$$\boldsymbol{\mathcal{S}} = \mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_{\boldsymbol{r}} \tag{1}$$

In n-gram LM, the probability of observing a sequence of words ($\mathcal{S}$) can be defined as follows:

$$\boldsymbol{P_n}\,(\boldsymbol{\mathcal{S}}) = \prod_{\boldsymbol{i=1}}^{\boldsymbol{r}} \boldsymbol{P}(\mathcal{K}_1|\mathcal{K}_{\mathbf{i-1}}, \mathcal{K}_{\mathbf{i}-2}, \ldots, \mathcal{K}_{\boldsymbol{i}-(\boldsymbol{n}-\mathbf{1})}) \tag{2}$$

where n is the order of the Markov process. This probability is dependent on pre-ordering the $n-1$ words that appear in a text. The zero-gram LM is when $n = 0$. In this case, the probability of any word has the same value. However, when $n = 1$, it is called uni-gram LM. The probability of any word is dependent on its frequency on the text. We use the uni-gram LM in this research.

3 Methodology

In this section, the new Arabic OCR database is presented. After that, a new alignment algorithm is proposed, called Fast Automatic Hashing Text Alignment (FAHTA). This algorithm is used to align the misrecognized word forms with the corresponding ground truth word forms. Then, the Arabic

context-dependent EM using WFSTs and the statistical language model are used to generate the correction rules. Finally, the two Arabic OCR post-processors are implemented and compared.

3.1 Preparing Arabic OCR dataset

There is no large Arabic OCR context database published to be used by the research community. According to [22,44], the best size of OCR dataset must reach 9000 images for any language. In addition, the OCR database may be divided into two parts, training and testing. The training size is suggested to be 6000 images and the testing size is 3000 images.

We developed an Arabic OCR database that contains the ground truth dataset and their scanned images. The scanned images are converted into text using ABBYY OCR software. We extracted 40,000 HTML files randomly from the Arabic Wikipedia site. The HTML files represent the ground truth database. The extracted files contain rich text Arabic articles. The images, tables, and other non-textual contents are removed. Then, the HTML files are sorted based on their sizes. From the set of extracted HTML files, the 4581 largest size files were selected to be scanned. The files are printed and scanned using RICOH Aficio MP 7500 scanner. The scanned files are saved in the scanned database. Each scanned file was named with the same corresponding golden file.

We convert the HTML file into a PDF file by saving the HTML file as a PDF file with A4 page size using the default margin setting in the Google Chrome Web browser. The PDF files were printed and the printed sheets were scanned. Each file in the scanned image database is converted into text using ABBYY OCR software. The OCR returns the output as a text file.

A single image contains a maximum number of 800 words and a maximum number of 40 lines. The average number of words in a single image is 491, and the average number of characters in a single image is 3000. The *Wikipedia* articles have different font types, boldness, size, and style. The text may contain different styles like bold, italic and underline. The document font size can be between 14 to 40px. All golden files were extracted in the utf-8 coding.

According to [22], the standard size of OCR document dataset has to be around 9000 sheets for any language. Hence, the size of our proposed database is 8994 images.

3.2 Language model

The language model (LM) is generated using the Arabic corpus using the set of words along with their frequencies. LM is built using a hash table where the weight of each word is extracted from the frequencies calculated using the corpus. In order to construct the uni-gram model, we implement the hash table that includes each word in the corpus with its probability estimates. We apply Eq. (3) to calculate the probability $p(w_i)$ of observing the **i**th word in the corpus [11]:

$$p(w_i) = \frac{count(w_i)}{count(W)} \tag{3}$$

where w_i is the **i**th word in the corpus, W is the corpus size, $count(w_i)$ is the frequency value of the ith word, and $count(W)$ is the number of words in the corpus.

For the correction stage, the LM is constructed as a hash table. This makes the word search faster and more efficient. The LM may be unable to find the best suggestion for the unseen word forms. Therefore, the error model is used to generate a set of suggested corrections for the words with errors [12].

3.3 A fast automatic hashing text alignment for documents (FAHTA)

The alignment technique is used to find the matching between the golden text and the misrecognized text. In this research, the alignment technique is utilized for two purposes. First, it is used to find the ground truth words that match the misrecognized words. Second, it is used to automatically evaluate the OCR results before and after applying the OCR post-processing.

The alignment algorithm starts by matching the golden database against the document that we are trying to correct. The golden database is the set of HTML files extracted from Wikipedia. The document we are trying to correct is the text file generated after converting the scanned image into text using the OCR software.

In this study, a new alignment algorithm is proposed. We call the proposed algorithm fast automatic hashing text alignment (FAHTA). The algorithm steps are shown in the following pseudo-code:

Input:*the golden string and the misrecognized string*
Output:*pairs of ground truth token and the corresponding misrecognized token.*
Steps:

1. *Find the similarity measure between each pair of inputs*
2. *Define the difference set as hash set of the golden string.*
3. *Remove the common tokens (anchor tokens) from the difference set*
   a. *Find the closest token in the misrecognized set by applying Levenshtein Distance algorithm.*
   b. *Return pair of the ground truth token and the corresponding misrecognized token.*
4. *Compute the error rate for misrecognized string.*

The proposed algorithm starts by computing a similarity measure between the golden string and the misrecognized string.

The similarity measure was proposed by [17] as shown in Eq. 4. If $W(d_i)$ is the set of all unique words inthe document d_i, then the resemblance measure $R(d_i, d_j)$ between document d_i and document d_j is defined as follows:

$$R(d_i, d_j) = \frac{|W(d_i) \cap W(d_j)|}{|W(d_i) \cup W(d_j)|}. \quad (4)$$

This equation computes the similarity value between the documents in an efficient and fast way. Hence, if the similarity value equals to 1, then this means that the tested documents are similar. Thus, there is no need to continue (i.e., this step takes $O(\text{Min}(|d_i|, |d_j|))$. Otherwise, each document must be divided into substrings (i.e., this step takes $O(|d_i|)$).

After obtaining the substrings, the intersection set of tokens between the tested documents is extracted (i.e., this step takes $O(\text{Min}(|d_i|, |d_j|))$. The difference set is initially defined as the hash set from the golden set. Then, the intersection set is removed from the difference set (DS).

Finally, to return the ground truth token and the corresponding misrecognized token, each token in the difference set is searched in the set of misrecognized tokens by using an edit Levenshtein distance algorithm [30] (i.e., this step takes in the worst case $O(|\text{DS}/2|^2)$). Figure 3 shows an example of applying the algorithm on an Arabic text.

The novelty of this algorithm comes from using the concept of the anchor word. The anchor words are the unique tokens in the misrecognized text which are not found in the ground truth. Hence, there is no need to use a recursive stage as in [46,47]. The similar words found when we compare between the misrecognized text and the ground truth are then used to find the misrecognized text.

Previous algorithms compute the similarity between two texts and return the common pairs of words in the original order. However, they do not return the pairs of the ground truth and the misrecognized word (i.e., the difference set). However, our proposed algorithm can return this pair.

3.4 Context-dependent confusion rules extraction

The purpose of the EM is to speed up the correction process for OCR errors by suggesting the most suitable word to be used to replace the incorrect word. Al Azawi and Breuel [12] proposed an EM that uses context-dependent confusion rules.

A context-dependent confusion rule is a two-sided rule that was extracted from the output of the alignment method. The left side of the rule represents the extracted substring from the misrecognized word, and the right side of the rule represents the extracted substring from the corresponding ground truth word. This model is built using Levenshtein edit distance algorithm. A number of edit distance operations are applied to correct the misrecognized word.

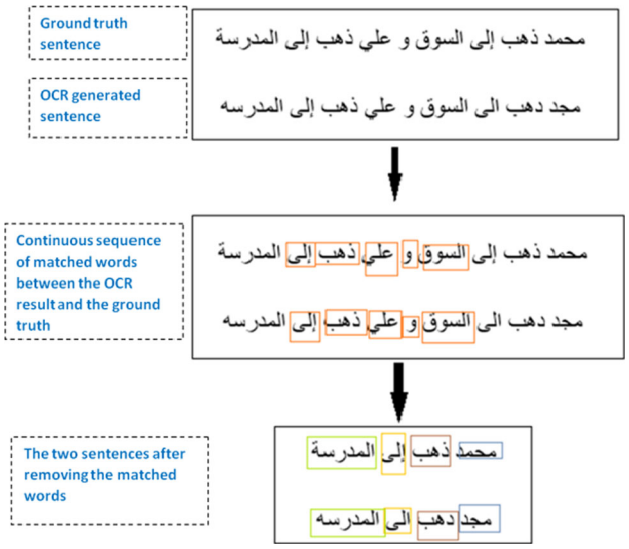

Fig. 3 Example of applying the FAHTA algorithm

Table 1 Example of context-dependent rules generated to be used for correcting misrecognized words

Context Dependent Rules	Misrecognized Words	Correct Words
عيد → عهد	عد (end)	عيد (eyd)
تين → بهن	بن (bn) (the son of)	تين (tyn)
على → علي	علي (Ali)	على (ala) (on)

The EM assigns a cost for each rule. The cost is represented using the weight which is assigned according to the probability of the OCR output word ***r***. The weight is assigned to the pair of the misrecognized word ***r*** and its correct word ***s***. Examples of context-dependent rules are shown on Table 1.

3.5 Arabic OCR post-processing framework

The OCR post-processing stage is used to detect and correct the misrecognized words generated by the OCR. In order to correct the OCR errors, the implemented EM and LM are used. In this paper, two OCR post-processing frameworks are implemented. The first is a rule-based framework adapted from [12]. The second framework is our proposed hybrid framework. The two implemented OCR post-processing frameworks consist of the following two components:

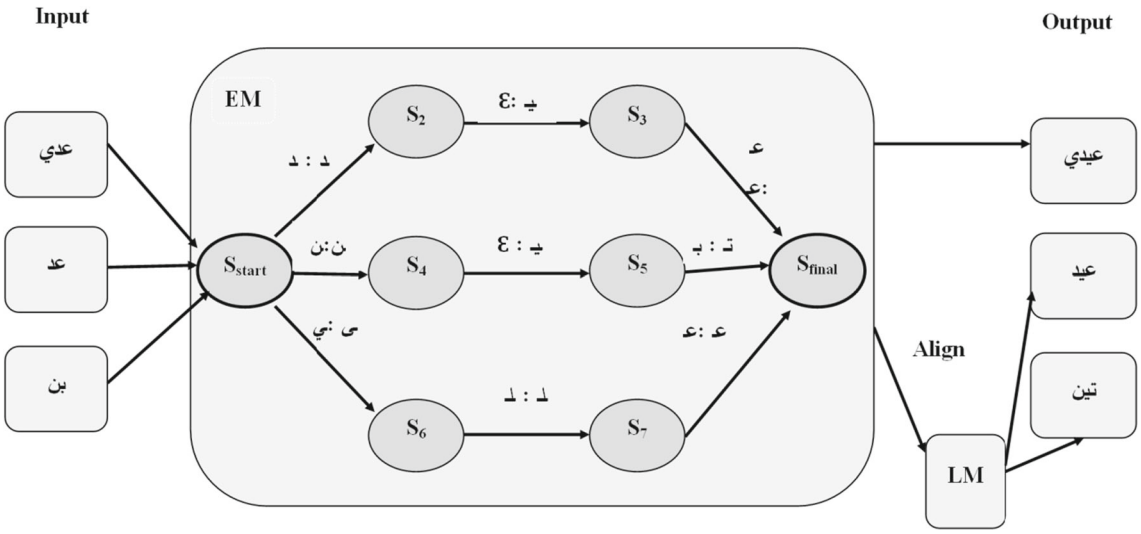

Fig. 4 Sample of the extracted rules in error model (EM)

1. LM: The language model is composed of a list of word forms along with their frequencies, which extracted from the training corpus. A two-sided hash table structure is used to represent the LM.
2. EM: The error model is composed of a list of the confusion rules along with their frequencies, which are extracted from the training dataset. A two-sided WFSM structure is used to represent the EM.

3.6 Offline Arabic OCR post-processing framework

Initially, the Arabic OCR post-processor takes the recognized document as input and returns the corrected document as output. For each document, we read each token sequentially and search for it in the LM. The selected token can be one of the following two cases: Either the token is found, and it is in a valid word form or the token is not found, and it is not in a valid word form.

In order to fix a token that is not in a valid word form, it must be parsed using the EM. The parsing will result in suggesting all the possible word forms. The found suggestion has two cases. First, the word form is not found in the EM. For this case, the token with the highest probability rule is used as the suggested correction. Second, the suggested set has more than one valid word form. In this case, the word form with the highest frequency is selected. An example of the correction scenario is shown in Fig. 4.

The EM can suggest the correct word form by matching the error token with the input label of the matched rule. Then, the output label of the matched rule is used to replace the error token (i.e., the corresponding input label) to change the token into the correct word form. The EM is represented using the WFST. Each rule represents the suggested path in the finite-state machine (FST). The rule starts with the $\mathcal{S}_{\text{start}}$ state and ends with the $\mathcal{S}_{\text{final}}$ state. The transition between any two states represents a single edit distance operation. For instance, the confusion rule عيد → عءد in the transducer shown in Fig. 4 represents the transition between $\mathcal{S}_2$ and $\mathcal{S}_3$ states as ε→ء. This rule represents an insertion operation.

3.7 Hybrid Arabic OCR post-processing framework

Bassil and Alwani [15] proposed OCR post-processing algorithm using Google's online spelling suggestion. Google's online spelling suggestion is based on the probabilistic n-gram model for predicting the next word in a particular sequence of words. Google's online spelling suggestion can suggest an alternative correct spelling for different errors such as: often made typos, misspellings, and keyboarding errors. These errors differ from the errors made by Arabic OCR.

Google's algorithm checks the spelling of each word in the search query and matches the query with Google's index database. If the occurrence of the query words is high, then no need for correction. However, if the occurrence of query words is low, then Google gives different suggestions using probabilistic *n*-gram which is based on the index database in the form of "*did you mean: spelling-suggestion*". For example, searching for "جهاز الايباج على الطاولة" (jehaza-libaj ala altawelah) Google's engine suggest *did you mean:* "جهاز الايباد على الطاولة" (jehaz Alipad ala altawlah)

The Google online spelling algorithm cannot be used alone for OCR post-processing. Hence, we hybridize the above method to enhance Arabic OCR post-processor results. The proposed framework for the hybrid system is shown in Fig. 5.

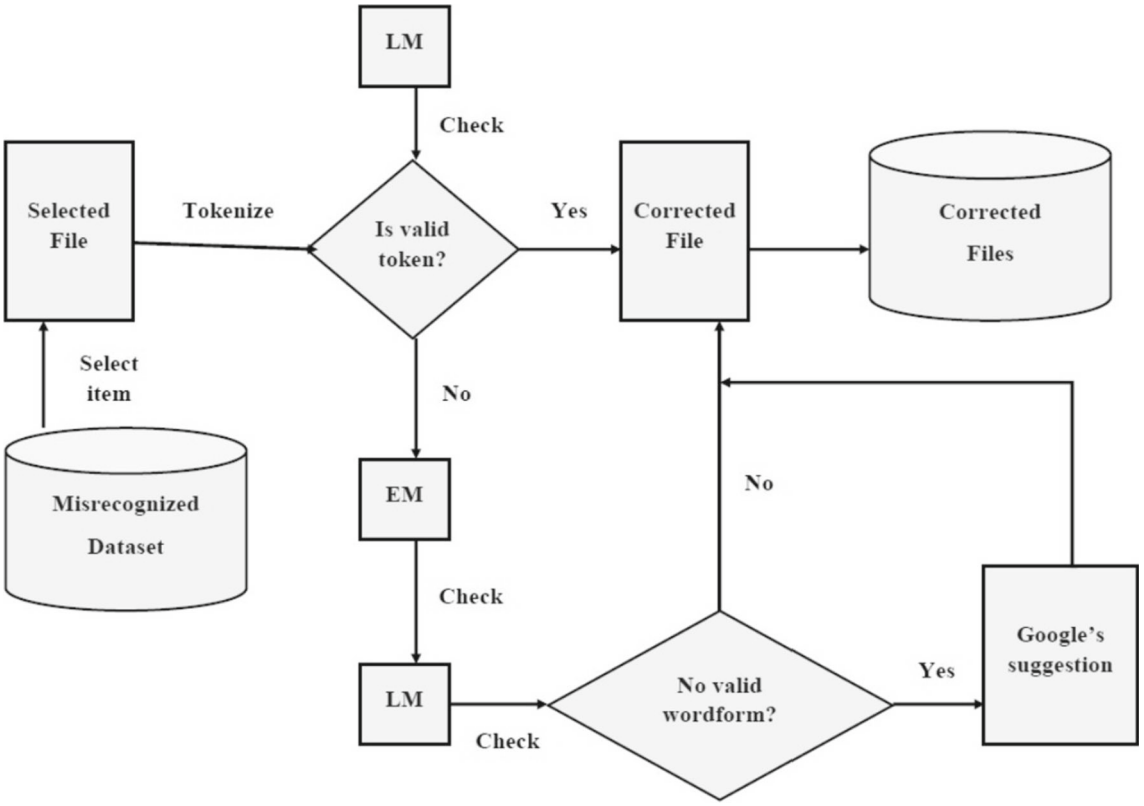

Fig. 5 The proposed hybrid Arabic OCR post-processing system

Experimentally, we find that using a 3-g query (i.e., the misrecognized word, the word before, and the word after) makes Google's suggestion results more accurate.

The OCR database (i.e., the OCR output and the corresponding Ground truth) are used for offline learning as the proposed hybrid algorithm generates the LM and the EM to be used by the OCR post-processing system. As Fig. 5 shows the proposed hybrid system starts by tokenizing the selected file (i.e., the file generated by the OCR). Each token (word) is checked against the LM to see if it is a valid token. If the token (word) is not found in the LM, then we use the EM to correct the word into a corrected word. After that, we look for the corrected word in the LM to check if it is a valid token. If the corrected word is not found in the LM (i.e., not a valid word form), then we take the word before and the word after (3-g) as a query.

The obtained 3-g query is used to search online using Google, and we use the Google's suggestion to correct the word. This process is repeated for all the tokens in the selected file and the system generates the corrected file which contains all the tokens after correction.

An example of error model rule is تنل → تقل, such rule can be used to correct the word المستنلة into المستقلة by applying a substitution operation on the character ـنـ to become the character ـقـ. Another example is correcting the word عي to the word على using the rule عـ ء ي →على, which apply an insertion operation to insert the character ـنـ.

An example of a misrecognized word is when ـلـ. (amabad) is generated from the under-segmentation problem as it is presented as one word. To solve this problem, the hybrid system uses the Google's suggestion. The generated suggestion is two words أمابعد (ama bad) which is used to correct the word.

Table 2 shows a sample of 10 words that are corrected using Google's suggestion in the hybrid system and the reason why it cannot be corrected using the LM or the EM.

The rule-based system was not able to suggest the valid word form for the word وانمحيط. On the other hand, the hybrid system takes the وانمحيط الاطلسي (wanmhyt alatasi) query and then return الاطلسي وان المحيط (waanna almohyt alatlasi) and the ground truth is والمحيط الاطلسي (waalmohyt alatlasi). In case the document contains a large number of unseen words, the rule-based system will not be able to suggest the correct word form.

3.8 Implementation

The framework is implemented using the following software components: **First, the Web crawler** which is used to extract the HTML files from Wikipedia site and it is used to create the Arabic OCR database. We used Crawler4j which is a Java library that provides a simple interface for crawling the Web. **Second, the jsoup parser** which is a Java library used for extracting and manipulating data, using DOM, CSS, and jquery methods. **Third, the Google's toolkit** from which we use the Diff, Match and Patch libraries to apply the operations required for synchronizing the plain text. **Lastly, the language model and the error model** which are developed using Java.

Table 2 Sample of 10 words that has been corrected using Google suggestion

No.	Error word	Reason of the error	Suggested correction by Google
1	المونننوعه	Did not match any of the suggested solutions from LM	الموسوعة
2	اكرة	Did not match any of the suggested solutions from LM	الحرة
3	ببيلعب	Did not match any of the suggested solutions from LM	سيلعب
4	قعسافات	Did not match any of the suggested solutions from LM	بمسافات
5	يثتخدم	Did not match any of the suggested solutions from LM	يستخدم
6	تعتبر <محمد	The word does not match any word in the LM or EM.	يعتبر محمد
7	عللى تكون	The word does not match any word in the LM or EM.	عللي تكون
8	الساءحبن	More than one error in the word and could not find the matching EM.	السائحين
9	دجذب	The word does not match any word in the LM or EM.	وجذب
10	فييي <الملعب	Did not match any of the suggested solutions from LM	في <الملعب>

4 Experiments

4.1 Experiments setting

The rule-based and the hybrid OCR post-processing systems are evaluated using the Arabic OCR database that we developed. The OCR database is divided into training and testing sets. We split the dataset based on [22,44]. The training size is 6002 images (pages), and the number of files is 2924. Also, the testing size is 2994 images (pages), and the number of files is 1657.

The experiments are conducted using a machine with Intel (R) Core (TM) i3-405U processor with 4 GB RAM. The operating system is WINDOWS 8 (64 bits).

The ground truth is used to evaluate the OCR post-processing correction. The rules are extracted from the training dataset. The total numbers of rules are 323,789. The uni-gram LM consists of 341,674 entries which are generated from the training dataset.

4.2 Experimental results

In the experiments, the OCR output and the corresponding ground truth are used to build the EM. Table 3 shows a sample of the generated context-dependent rules using the misrecognized words. The table shows the misrecognized word, the suggested word and the used edit distance operation to correct the word.

The word error rate (WER) is used to find the number of error words using the equation:

$$\mathrm{WER} = \frac{m}{c} \tag{5}$$

where m is the total number of matching tokens in the alignment (i.e., the matching tokens of the ground truth against the corrected text) and c is the total number of tokens in the golden file.

Table 3 Examples of the generated context-dependent rules using the misrecognized words

Context Dependent Rules	Misrecognized Words	Suggested Words	Edit Operations
تقل → تنل	المستنلة almstnlt	المستقلة Almostaqylt: the independent	Substitution
ت ع→ تا	مديراتا mdyrata	مديرات mdyrat: directories	Deletion
تين→ بعبن	بن bn	تين Tyn: fig	Insertion
خيط→حيط	وانمحيط wanmhyt	وانمخيط waalmkyt	No match

The two Arabic post-processing OCR systems are implemented. The first is the rule-based system adapted from [12]. The second system is the proposed Hybrid Arabic Text Error Detection and Correction system. The two systems can perform autodetection and correction of the OCR errors, and they are language independent and non-constrained by the string length. The proposed Arabic corpus is used to build the uni-gram statistical LM.

The average error rate for the rule-based system and the proposed hybrid system is presented in Table 4. The training sample represents a random sample taken from the training dataset which is used by the system to build the system learning. The testing sample represents a random sample taken from the testing dataset that is not used to build the system learning. The full sample represents the whole dataset including the training and testing sample.

Using a training sample of 1000 images the hybrid OCR post-processing system improves the results from an error rate of 24.02% to an error rate 18.96%. It performs better

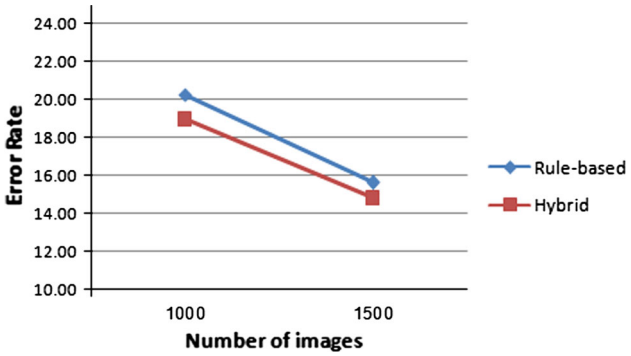

Fig. 6 The growth of error rate

than the state-of-the-art rule-based system which returns an error rate of 20.26% as shown in Table 4. In general, the results show that the hybrid system reduces the error rate in both training and testing samples.

We can see clearly that rule-based system has lower error rate when compared to the OCR result in training, testing, and full sample. The training dataset presents a larger enhancement than the testing or the full sample because it has a low number of unseen words. The testing dataset contains a larger number of unseen words compared to the training or the full sample which result in a higher error rate.

Figure 6 shows the growth of error rate in the rule-based system and the proposed hybrid system. We can see that the hybrid system has lower error rate than the rule-based system.

The hybrid system outperforms the rule-based system as it handles some of the cases that are not easy to be corrected. For example, if the word misses some letters, then the rule-based system cannot suggest the valid word form for the misrecognized word (e.g., "وانمحيط"). On the other hand, the hybrid system takes the word with error and the next word as الاطلسي وانمحيط (wanmhyt alatasi) and returns الأطلسي وان المحيط (waanna almohyt alatlasi) as a suggested correction. On the other hand, the golden text has this word as والمحيط الأطلسي (w aalmohyt alatlasi).

Table 5 shows three common types of errors which are over-segmentation (adding letters to the word), under-segmentation (removing letters from the word), and merging two words in one word. In case the word is in the LM or EM, then the system can fix the problem if the word has over-/under-segmentation errors. Nonetheless, if the word is not seen before, then the system will use Google's suggestion. On the other hand, if the word is generated because of merging two words into one word, then the proposed system relies on using Google's suggestion.

5 Conclusion and future work

This paper has achieved several key contributions: developing an Arabic document database that can be used for

Table 4 Average error rate for the rule-based system and hybrid system compared to the original OCR recognition result

Dataset	OCR	Rule-based system	Hybrid system
Training sample (1000 images)	24.02	20.26	18.96
Testing sample (500 images)	14.95	14.53	14.42
Full sample (1500 images)	20.99	18.35	17.44

Table 5 Sample of eight words with over-/under-segmentation and merge words errors

No.	Error word	Hybrid Arabic OCR	Golden	Error type
1	المونننوعه	الموسوعة	الموسوعه	Over-segmentation
2	ببيلعب	سيلعب	سيلعب	Over-segmentation
3	قعسافات	بمسافات	بمسافات	Over-segmentation
4	يثتخدم	يستخدم	تستخدم	Under-segmentation
5	اكرة	الحرة	الحرة	Under-segmentation
6	الموسوعهالحرة	الموسوعة الحرة	الموسوعه الحرة	Merge two words
7	فيالمكتبه	في المكتبة	في المكتبه	Merge two words
8	دربن	تدريب	درس	Merge two words

evaluating OCR systems, providing a novel alignment technique, and providing a novel Arabic OCR post-processing technique. The main goal of this research is to build a general-purpose language model method using a high-performance technique that is language independent and can work with any word length.

The Arabic OCR database is created along with its ground truth database. Our database has been classified into three groups: the HTML files, the PDF files, and the scanned images files. The HTML files represent the ground truth database from which the documents have been extracted randomly from Wikipedia Web site. The PDF files are the converted HTML files into a PDF format. Finally, the scanned image files represent the Arabic document OCR database which is scanned from the printed PDF files. The total number of files is 4581 and the total number of images (pages) is 8,994. This database is not classified according to a specific topic which is the future work of this study.

We proposed Fast Automatic Hashing Text Alignment for Documents (FAHTA) algorithm to evaluate the OCR accuracy of the scanned documents. The algorithm removes the intersection between the golden document and the corrected document. This approach can be used for other languages. The OCR suffers from under- and over-segmentation at the line and character level. However, our proposed approach uses the word level segment to avoid this problem. The proposed algorithm is affected by under- and over-word segmentation problems which will be investigated in our future work.

Another goal of this paper is also to develop a statistical Arabic language model based on hybridizing the EM approach with the context approach. We implemented the two Arabic OCR post-processing systems: rule-based system and hybrid system.

The experiments show that our proposed hybrid approach performs better than the state-of-the-art rule-based approach [12]. Also, our experimental results show that the rule-based system for the training dataset using 1000 images improves the word error rate from 24.02% to an error rate 20.26%. After training, the rule-based system uses 500 testing dataset images the results show that the word error rate enhanced from 14.95% to become 14.53%. In addition, the experimental results show that the hybrid OCR post-processing system improves the results of training sample using 1000 images from an error rate of 24.02% to an error rate of 18.96%. The hybrid OCR post-processing system improves the results of the testing sample from an error rate of 14.95% to an error rate of 14.42%. As shown in the results, the proposed hybrid system outperforms the rule-based system using a sample of 1500 images.

In the future, we plan to develop a statistical language model using a higher n-gram. Such system can be used for recognizing Arabic handwritten documents. We plan to create Arabic historical book OCR database to evaluate the proposed approach on low-quality images. Finally, we can apply the proposed algorithm when the Arabic text has diacritic marks.

References

1. Abdelraouf, A., Higgins, C.A., Khalil, M.: A database for Arabic printed character recognition. In: A database for Arabic printed character recognition, pp. 567–578. Springer, Berlin (2008)
2. Abdelraouf, A., Higgins, C.A., Pridmore, T., Khalil, M.: Building a multi-modal Arabic corpus (MMAC). Int. J. Doc. Anal. Recognit. (IJDAR) **13**(4), 285–302 (2010)
3. Abu Doush, I., Al-Trad, A.: Improving post-processing optical character recognition (OCR) documents with Arabic language using spelling error detection and correction. Int. J. Reason.-Based Intell. Syst. **8**(4), 91–103 (2015)

4. Abu Doush, I., Alkhateeb, F., Al Raoof'bsoul, A.: Semi-automatic generation of Arabic digital talking books. In: 2014 3rd International Conference on User Science and Engineering (i-USEr)
5. Abu Doush, I., Alkhatib, F., Bsoul, A.A.R.: What we have and what is needed, how to evaluate Arabic Speech Synthesizer? Int. J. Speech Technol. **19**(2), 415–432 (2016)
6. Alginahi, Y.M.: A survey on Arabic character segmentation. Int. J. Doc. Anal. Recognit. (IJDAR) **16**, 105–126 (2013)
7. Alkhateeb, F., Abu Doush, I., Albsoul, A.: Arabic optical character recognition software: a review. Pattern Recognit. Image Anal. **27**(4), 763–776 (2017)
8. Alkoffash, M.S., Bawaneh, M.J., Muaidi, H., Alqrainy, S., Alzghool, M.: A survey of digital image processing techniques in character recognition. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) **14**(3), 65 (2014)
9. Amin, A.: Segmentation of printed Arabic text. In: Advances in Pattern Recognition—ICAPR 2001. Springer, Berlin, pp. 115–126 (2001)
10. Amin, A., Masini, G.: Machine recognition of multifont printed Arabic texts. In: Proceedings of International Conference on Pattern Recognition, Paris, France, pp. 392–395 (1986)
11. Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F., Purdy, D., Smith, N., Yarowsky, D.: Statistical machine translation. Final Report, JHU Summer Workshop, p. 30 (1999)
12. Al Azawi, M., Breuel, T. M.: Context-dependent confusions rules for building error model using weighted finite state transducers for OCR post-processing. In: 11th IAPR International Workshop on Document Analysis Systems, pp. 116–120 (2014)
13. Al Azawi, M., Hasan, A. U., Liwicki, M., Breuel, T. M.: Character-level alignment using WFST and LSTM for post-processing in multi-script recognition systems-a comparative study. In: Image Analysis and Recognition. Springer, Berlin, pp. 379–386 (2014)
14. Al Azawi, M., Liwicki, M., Breuel, T. M.: WFST-based ground truth alignment for difficult historical documents with text modification and layout variations. In: IS&T/SPIE Electronic Imaging, vol. 8658, pp. 18-865818-12 (2013)
15. Bassil, Y., Alwani, M.: Ocr post-processing error correction algorithm using google online spelling suggestion (2012). arXiv preprint arXiv:1204.0191
16. Beaufort, R., Mancas-Thillou, C.: A weighted finite-state framework for correcting errors in natural scene OCR. Ninth Int. Conf. Doc. Anal. Recognit. **2**, 889–893 (2007)
17. Broder, A.Z.: On the resemblance and containment of documents. In: Compression and Complexity of Sequences Proceedings, pp. 21–29 (1997)
18. Broumandnia, A., Shanbehzadeh, J., Nourani, M.: Segmentation of printed Farsi/Arabic words. In: IEEE/ACS International Conference on Computer Systems and Applications, AICCSA'07, pp. 761–766 (2007)
19. Chang, J.J., Chen, S.-D.: The postprocessing of optical character recognition based on statistical noisy channel and language model. In: Proceedings of PACLIC, pp. 127–132 (1995)
20. Dađason, J.F.: *Post-correction of Icelandic OCR text.* Master's thesis, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland (2012)
21. Gharaibeh, A.: *A Hybrid Approach for Arabic OCR Post-Processing Using Rule Based and Word Context Techniques*, Master Thesis, Yarmouk University (2016)
22. Guyon, I., Haralick, R.M., Hull, J.J., Phillips, I.T.: Data sets for OCR and document image understanding research. In: In Proceedings of the SPIE-Document Recognition IV, pp. 779–799 (1997)
23. Habeeb, I.Q., Yusof, S.A., Ahmad, F.B.: Two bigrams based language model for auto correction of Arabic OCR errors. Int. J. Digit. Content Technol. Appl. **8**(1), 72 (2014)
24. Hall, P.A., Dowling, G.R.: Approximate string matching. ACM Comput. Surv. (CSUR) **12**(4), 381–402 (1980)
25. Kalt, T.: A new probabilistic model of text classification and retrieval. Technical Report IR-78. Citeseer (1996)
26. Kanoun, S., Slimane, F., Guesmi, H., Ingold, R., Alimi, A. M., Hennebert, J.: Affixal approach versus analytical approach for off-line Arabic decomposable vocabulary recognition. In: 10th International Conference on Document Analysis and Recognition (ICDAR'09), pp. 661–665 (2009)
27. Khorsheed, M.S.: Off-line Arabic character recognition-a review. Pattern Anal. Appl. **5**(1), 31–45 (2002)
28. Kukich, K.: Techniques for automatically correcting words in text. ACM Comput. Surv. (CSUR) **24**(4), 377–439 (1992)
29. Lee, Y.-S., Papineni, K., Roukos, S., Emam, O., Hassan, H.: Language model based Arabic word segmentation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, pp. 399–406 (2003)
30. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Phys Dokl. **10**, 707–710 (1966)
31. Liu, X., Croft, W.B.: Statistical language modeling for information retrieval. DTIC Document (2005)
32. Llobet, R., Navarro-Cerdan, J.R., Perez-Cortes, J.-C., Arlandis, J.: Efficient OCR post-processing combining language, hypothesis and error models. In: Structural, Syntactic, and Statistical Pattern Recognition. Springer, Berlin, pp. 728–737 (2010)
33. Magdy, W., Darwish, K.: Arabic OCR error correction using character segment correction, language modeling, and shallow morphology. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 408–414 (2006)
34. Magdy, W., Darwish, K.: Effect of OCR error correction on Arabic retrieval. Inf. Retr. **11**(5), 405–425 (2008)
35. Mostafa, M.G.: An adaptive algorithm for the automatic segmentation of printed Arabic text. In: 17th National Computer Conference, pp. 437–444 (2004)
36. Najoua, B.A., Noureddine, E.: A robust approach for Arabic printed character segmentation. Proc. Third Int. Conf. Doc. Anal. Recognit. **2**, 865–868 (1995a)
37. Nayak, M., Nayak, A.K.: Odia running text recognition using moment-based feature extraction and mean distance classification technique. In: Intelligent Computing, Communication and Devices, Springer (2015)
38. Saad, R., Elanwar, R., Abdel Kader, N., Mashali, S., Betke, M.: BCE-Arabic-v1 dataset: towards interpreting Arabic document images for people with visual impairments. In: PETRA '16, Corfu Island, Greece (2016)
39. Schlosser, S.: ERIM Arabic Database. Environmental Research Institute of Michigan, Ann ARbor (2002)
40. Schulz, K.U., Mihov, S.: Fast string correction with Levenshtein automata. Int. J. Doc. Anal. Recognit. **5**(1), 67–85 (2002)
41. Slimane, F., Ingold, R., Kanoun, S., Alimi, A.M., Hennebert, J.: Database and Evaluation Protocols for Arabic Printed Text Recognition. DIUF-University of Fribourg, Switzerland (2009)
42. Slimane, F., Kanoun, S., El Abed, H., Alimi, A. M., Ingold, R., Hennebert, J.: ICDAR2013 competition on multi-font and multi-size digitally represented arabic text. In: 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 1433–1437 (2013)
43. Toselli, A.H., Romero, V., Vidal, E.: Alignment between text images and their transcripts for handwritten documents. In: Language Technology for Cultural Heritage, Springer, Berlin (2011)
44. Ul-Hasan, A., Bin Ahmed, S., Rashid, F., Shafait, F., Breuel, T. M.: Offline printed Urdu Nastaleeq script recognition with bidirectional LSTM networks. In: 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 1061–1065 (2013)
45. Wemhoener, D., Yalniz, I.Z., Manmatha, R.: Creating an improved version using noisy OCR from multiple editions. In: 12th Inter-

national Conference on Document Analysis and Recognition (ICDAR), pp. 160–164 (2013)
46. Yalniz, I.Z.: Efficient representation and matching of texts and images in scanned book collections. Doctoral Dissertations in University of Massachusetts (2014)
47. Yalniz, I.Z., Manmatha, R.: A fast alignment scheme for automatic ocr evaluation of books. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 754–758 (2011)