Indonesian Journal of Electrical Engineering and Computer Science
Vol. 21, No. 1, January 2021, pp. 233~241
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v21.i1.pp233-241

Corpus-based technique for improving Arabic OCR system

Ahmed Hussain Aliwy[1], Basheer Al-Sadawi[2]
[1]Faculty of CS and Mathematics Department of Computer Science, University of Kufa, Kufa, Iraq
[2]Information Technology Research and Development Centre, University of Kufa University of Kufa, Kufa, Iraq

Article Info

Article history:

Received Apr 8, 2020
Revised Jun 24, 2020
Accepted Jul 8, 2020

Keywords:

AOCR post-processing
Arabic optical character recognition
N-gram language model
NLP-based OCR
Non-word error
Real-word error

ABSTRACT

An optical character recognition (OCR) refers to a process of converting the text document images into editable and searchable text. OCR process poses several challenges in particular in the Arabic language due to it has caused a high percentage of errors. In this paper, a method, to improve the outputs of the arabic optical character recognition (AOCR) Systems is suggested based on a statistical language model built from the available huge corpora. This method includes detecting and correcting non-word and real words error according to the context of the word in the sentence. The results show that the percentage of improvement in the results is up to (98%) as a new accuracy for AOCR output.

Corresponding Author:

Basheer Al-Sadawi
Information Technology Research and Development Centre
University of Kufa, Kufa, Iraq
Email: basheer.alasdi@uokufa.edu

1. INTRODUCTION

In the tech era, diverse techniques have been produced increasingly to access and acquisition of the information besides it leads to extend the revolution of science and knowledge that affect human life facilities and animations. These techs such as techniques of data mining, deep learning, AI, classifications, Natural Language processing and etc. [1, 2]. For several decades, newspaper articles, books, and researches have been digitized to make resources available to researchers and readers. The digitization process is done by optical character recognition system is branch of pattern recognition and artificial intelligence that convert the image of the text into an machine-readable text, which makes these texts usable to be processed by other tools or task such as indexing, machine translation, and search engine [3]. Optical character recognition is difficult task for many reasons such as low scanning and printing quality. thus, lead to no good result for text recognition. This difficulty is increased in case of recognition of a high inflected language such as Arabic language, due to the morphological and script characteristics of Arabic language [4]. As well as, Some AOCR systems produce multiple bad outputs for the same document image [5]. This is where natural language processing (NLP) that presented considerable progress in machine learning [6] to improve OCR output. In this paper, a post-processing system is suggested for improving AOCR output based on a language model (LM) built from huge corpora combined from several sources.

OCR process converts image of printed or handwritten text into digital text that can be modified, processed, searched, and copied [7]. Although there are still several shortcomings of the technology that need to be treated and resolved to raise accuracy, OCR is a mesmerizing technology that has shouldered computers to digitize texts [8]. The manipulation process includes segmentation image into lines, words and parts of words, thus segmenting the word image into images of characters. Character images are sequentially

recognized by classifiers to convert them into text encodings. The results of OCR systems may be not satisfactory when these systems work on low-quality documents such as old documents or these systems work according to incompetent segmentation algorithms and thus will lead to many non-recognized letter image which reduce the accuracy as a result of mismatch with samples.

There are two types of errors: (i) non-words errors and (ii) real word error. First, non-word errors are the words that cannot be considered valid words, because these words does not exist in the lexicon [9]. In this type of errors some letters are replaced with symbols or numbers. For example, OCR system recognize the words ("كتاب"-"book") and (يلعب""- "play") as "كت!اب" and "يلع7" respectively. This type of error easy to detect because the probability of a word containing symbols or numbers is very small or zero [10]. It may also appear when a letter of the original word is replaced by other letters, but the resulting is a non-valid word in the language. For example, the word ("يدرس" – "study") recognized as ("ظدرس"). The detection of these types of errors depends on the efficiency of the used dictionary.

Second, real-word (also called context-sensitive or semantic) error is a class of error that escapes the typical errors checker which based on dictionary look-up [11]. This type of error word is difficult to detect because the resulting word is another valid frequent word in the language but semantically or grammatically incorrect with respect to its context [10]. For example, case of OCR system recognizes the word ("صفقة" – "deal") as the word ("صنعة" – "craft") in the context ("صفقة القرن المشؤومة" – "The fateful deal of the century") where these words are very close in scripting. These error words cannot be handled separately to classify them as errors because the process of disclosure needs more contextual information. Context-based methods are used for detecting this type of error.

2. RELATED WORKS

Several systems and works have been proposed to improve the output of OCR system for Arabic language. The most common used OCR systems for Arabic language are Omnipage and Sakhr Automatic Reader that evaluated by Kanungo et al. (1999) [12]. Suzan Verberne (2002) [13] constructed a context-sensitive spell checking BESL by using the probabilities of trigram words to detect and correct real-word errors. Tahira Naseem (2004) [14] employed two methods for spell check of Urdu language (i) Soundex algorithm techniques and (ii) Single Edit Distance for spelling correction. Magdy and Darwish (2006) [4] introduced a post processing system consisting of three models to improve the output based on the Levenshtein edit distance model, trigram language model, and shallow morphological model. Shaalan et al. (2012) [15] proposed post-processing system for (i) detecting misspell words by direct detection and language model based detection, and (ii) correction model consists of generating candidates and selection by noisy channel and then minimum edit distance. Doush and Al-Trad (2016) [16] Developed AOCR post-processing system based three strategies Google online suggestion system, Ayaspell spell checker with Google online suggestion system and Microsoft Office Word with Google online suggestion system. Imad Qasim (2016) [17] proposed a hybrid model from combined three improved techniques of alignment, differentiation, and voting to overcome the identified drawbacks to recognize the optical characters in the Arabic language. Anwaar Hamdi (2016) [18] developed the statistical Arabic Language Model by hybridization context approach with Error Model approach to improving the output of AOCR systems. Sonia Yousf et al. (2017) [19] presented improving Long-Short Term Memory (LSTM) of AOCR of text in videos by recurrent connections language modelling by focusing on two factors Recurrent Neural Network (RNN) for language modelling and decoding schema. Doush, Alkhateeb, and Hamdi (2018) [20] Proposed model of language-independent a AOCR post-processing system by two frameworks the Language model and hybrid error model with contextual model.

3. ARCHITECTURE OF THE PROPOSED SYSTEM

In this paper, a post-processing of OCR outputs with natural language processing techniques is suggested. The proposed post-processing consists of two parts (i) the first part is the construction of a language model for massive data after collecting of them. (ii) The second part is the detection and correction of errors. The structure of proposed post-processing system is shown in Figure 1.

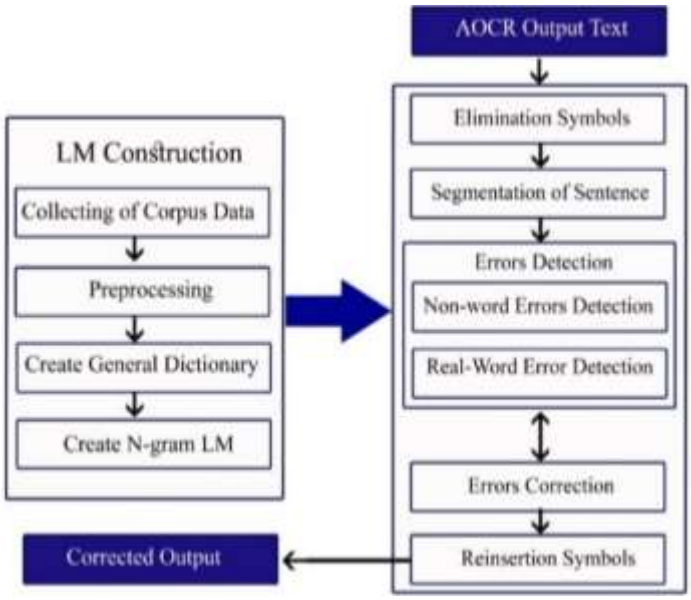

Figure 1. Structure of post-processing system

3.1. Construction of language model (LM)

The first part of the proposed system is construction of LM from huge Arabic texts. It consists of four steps; data collection, preprocessing, dictionary construction, and n-gram language model construction. These steps are described in the following sections.

3.1.1. Data collection

Huge texts were collected from many sources for construction a suitable corpus for our work. The collected texts were in different extensions (Doc, Docx, txt, and Pdf). Number of words in the collected corpora is 1,099,054,713 word in different domains. These sources are AL SHAMILA LIBRARY (994,011,955 words) collected from Islamic books, ANT Corpus v1.0 and v1.1 [21, 22] (1,474,000 words) collected from news articles of Tunisian web radio site Jawhara FM, ANT Corpus v2 [23] (9,670,000 words) from News articles and, AraCorpus (93,898,758 words) collected from News and essays articles.

3.1.2. Preprocessing

The pre-processing stage aims to reduce the noise in the combined text of Corpus files. This stage includes the following operations:

a) Tatweel removal: removing the symbol that make word more stretch [24]. (e.g. the word ("قـــــــال" – "say") become ("قال" – " say"))
b) Diacritics removal: removing special marks that appear with Arabic letters (e.g. the phrase (" فَإِنَّ لَكُم مَّا سَأَلْتُمْ") became ("فإن لكم ما سألتم").
c) Numbers removal: the elimination of numbers from the text such as dates, numbering and references numbers.
d) Foreign text removal: elimination of foreign scripts (non- Arabic scripts) from the text (e.g. the abbreviation BBC).
e) Special symbols removal: removing symbols that is not considered sentence boundaries such as (#, @,! , $,%,^,&,*,|,~,+,=).
f) Single letter removal: elimination of the single letters because the minimum length of a word in Arabic is two letters [25].

3.1.3. Dictionary construction

After reducing the noise from the corpus, the process of construction of the dictionary starts by tokenization of the running text into words depending on the white space. These words are stored in the dictionary as keys, while word frequencies in the Corpus are stored as the values for these keys. This is can be considered as unigram dictionary where it is based on one word.

3.1.4. Construction of N-gram language model

This phase involve the construction of the heart of the proposed system that represent the main part for detecting and correcting errors. This phase can be achieved by performing the following task:

a) Sentences Segmentation which can be achieved using punctuation marks such as { ? , [,] , (,) , . , }.
b) Adding boundaries tags (</s> and <s>) to each sentence.
c) Constructing N-gram models for N=2, 3, 4, 5 with their frequencies by Acquisition chains of words by moving the sliding window with a specific length N in one item (word) at a time. The text chain represents the words below the sliding window fields after the movement.

3.2. Errors detection and correction

The methodology, which used in this paper for correction misspelling errors, is achieved by generating correction's candidates and ranking of them. The proposed system detects and corrects the both types of errors non-word errors and Real-world errors by following the main steps that is shown in Figure 2.

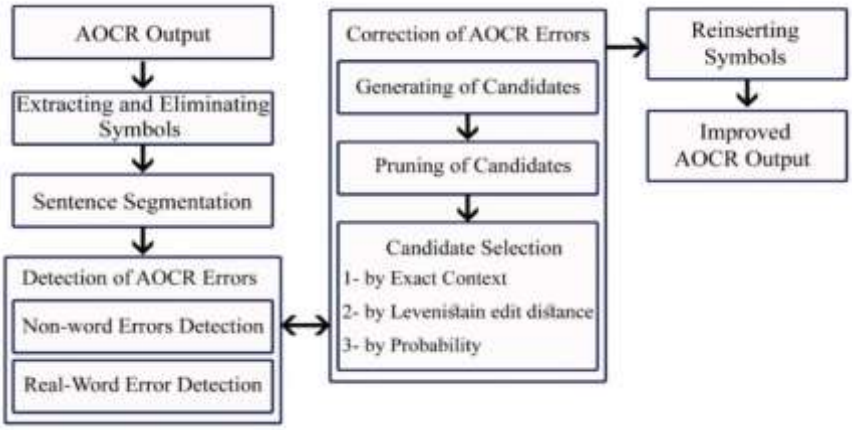

Figure 1. The steps of error detection and correction

3.2.1. Extracting and eliminating symbols

Any term (numbers, symbols, foreign characters, etc.) should be indicated with their indices in the text and if they are not exists in the punctuation list, they will be replaced by space in the text. After completion of the correction process, these symbols will be inserted to their original places. The separated Arabic letters are treated like these symbols because these letters can be as semantic symbols such as page number like ("ص 20").

3.2.2. Sentence segmentation

Sentence Segmentation is the process of splitting a text into sentences using punctuations symbols where the boundaries of each sentence is indicated by the symbol of begin and end (<s>, <\s>).

3.2.3. Detection of AOCR errors

In the proposed system, the non-word error is detected by looking in the unigram dictionary that constructed in the Language model construction phase to indicate whether a word is exists in the language vocabulary or not. If it does not exist in the dictionary, it considers Out Of Vocabulary word (OOV). The Real-word error detection is done by lookup for this word with its two adjacent words (one from left and one from right) as a chain in the trigram LM database that previously prepared. When this chain of words is exist in the trigram LM, this word is considered correct; otherwise, it is considered a Real error word. For example, the word ("السلاح" – "weapon") is correct, and it is present in the unigram dictionary, but if this word appears in the sentence ("حرث السلاح الارض"-"the weapon plowed the land"), it will be a real-word error. Because these three words not frequented in the trigram LM. Therefore, this word must be corrected to become ("حرث الفلاح الارض"- "The farmer plowed the land") as shown in Figure 3.

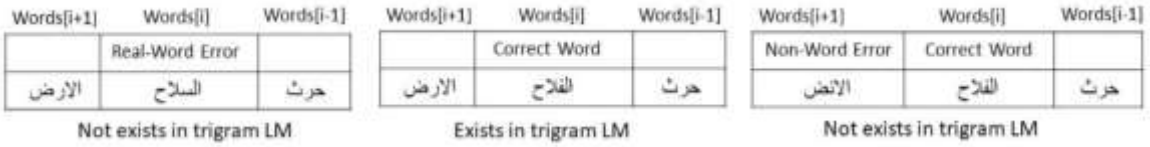

Figure 2. Detection of real-word error

3.2.4. Correction of AOCR errors

The corrections of both types of errors (real-word error and non-word error) are done using three main steps respectively (A) generating of candidates (B) pruning of candidates and (C) candidate selection.

1) Candidates generation

Firstly for each error word, the candidate's words should be produced. The process of generating candidates is a form of edit distance where the generator works basically as a character-based generator. It works by a set of operations that applied to error words to generate a list of all possible words within a single edit distance. These operations include generation by substitution, insertion, deletion and Pair Letters Substitution. Suppose that the length of error word is n, these steps are as following.

a) Generation by Substitution: It is the process of generating a list of candidates by replacing each letter in error word by each letter from the Arabic alphabet. The produced list will be 29n candidates, in addition three final letters (ء ,ى ,ة) to be (29n+3).
b) Generation by Insertion: It is the process of generating a list of candidates words by inserting each letter of Arabic alphabet after each letter in the error word or before the first letter in error word. The produced list will be (29(n+1) +3) candidates.
c) Generation by Deletion: It is the process of generating a list of candidates words by deleting each letter from error word one at a time. The remainder of the error word is added to the candidate's list. The produced list will be (n) candidates.
d) Generation by Pair Letters Substitution: It is the process of generating a list of candidates words by replacing every two adjacent letters in the error word with all letters of Arabic alphabet. The produced list will be (29(n-1) +3) candidate. This operation tries to correct the recognition of more than one letter errors.

2) Pruning of candidates

The candidates, generated by the previous operations, are considered as brute force operations so it produces a huge number of candidates. For example, when the length of error word is 6 characters long, the total candidates will be 537 as shown in Table 1. Because there are large numbers of generated candidates, filtering and elimination of incorrect candidates should be applied using the dictionary lookup in this stage.

Table 1. The generated candidates count for a word with length 6 letter

Operation	Formula	Candidate's Number
Substitution	29n+3	177
Insertion	29(n+1)+3	206
Deletion	n	6
Pair-Substitution	29(n-1)+3	148
Total		537

3) Candidate selection

After the candidates generation and prunetion, one of these candidates should be selected to be the correct alternative word for the error word. In the proposed system, three methods are used and applied sequentially for selecting the alternate candidate: (1) Selection by Exact Context, (2) selection by edit distance, and (3) selection by probability.

- Selection by exact context

Choosing the correct word among candidates is affected by the context. This feature is adopted in the language model that built from the Corpus. suppose that the position of the error word is (e) and the candidate list for the error word is (Ce). For all the words in positions (e+1, e+2, e+3 and e+4), the candidate lists (Ce+1 ,Ce+2 ,Ce+3 and Ce+4) will be generated respectively if any of these words is detected as error word. Offcourse if any word in these positions is correct word, C will has one word (this word). From these lists, the combinations of 2-gram,3-gram,4-gram and 5-gram will be produced with checking them validity in n-gram LM database. The result chains (ngrams) will be taken if exists in 5-gram,4-gram or 3-gram

combinations lists respectively. The details of this description are shown in Figure 4. if this mechanism cannot make a decision, the same scenario will be done on the words on right of error word at positions (e-1,e-2, e-3 and e-4). For example, suppose that the output of AOCR that contains errors is the word sequence ('جال لي ممل الذي عميه') where the error word to be corrected by context is ('جال'). it is corrected by the phrase neighbours to it ('الذي‘ ,'ممل‘ ,'لي' and 'عميه'). According to the five lists of cndidates and their combination for 5-grams, there are three 5-grams (chains) exist in the 5-gram language model therefore they will be the output of the context-based correction as shown in Figure 5. As we can see the other combination for 4, 3 and 2 grams will be neglected.

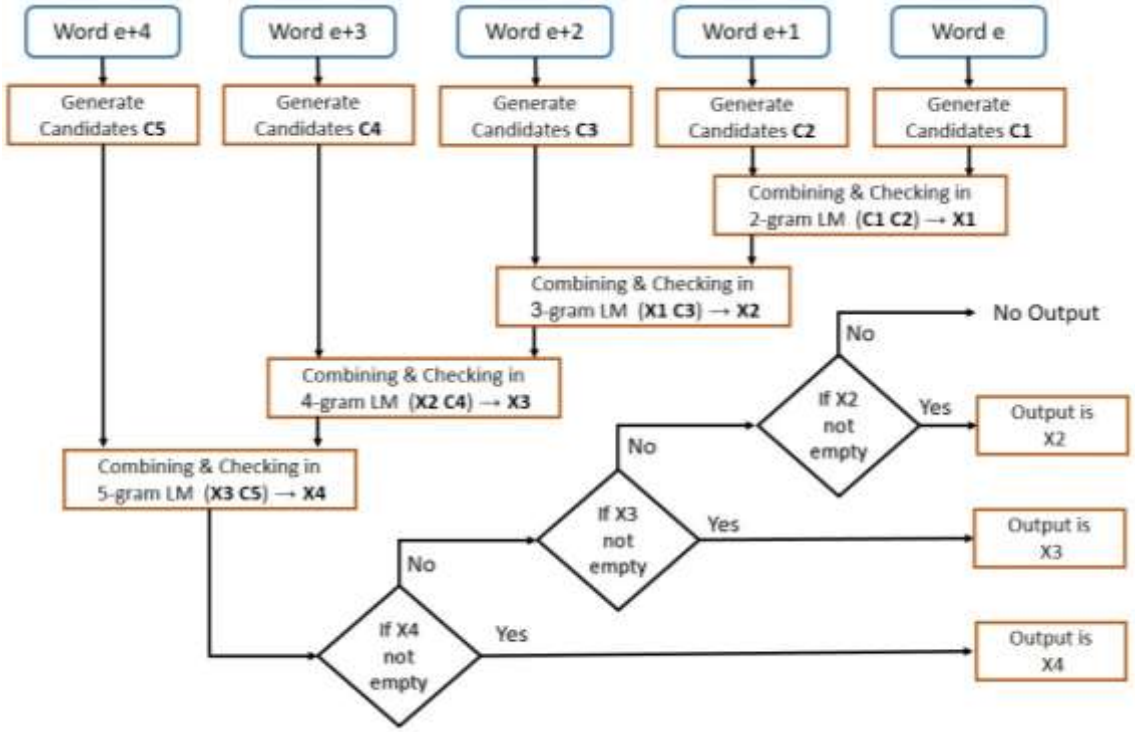

Figure 3. The steps of selecting the best candidates, according to the context

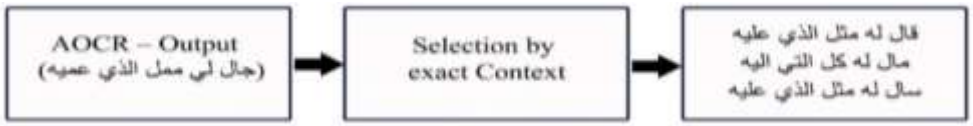

Figure 4. Example of selection candidate by exact context

- Selection by levenshtein edit distance measure

The output of selection by exact context will be a list of 5-gram, 4-gram or 3-gram chains. If the list contains more than one chain, one of them should be selected as the optimal chain that contains the correct word for the selected error word. The decision can be made by calculating minimum edit distance between the chains and the equivalent sequence of words in AOCR output. After completing the calculation of the edit distance for all the chains, the chains with the minimum edit distance are adopted, taking into consideration that there may be more than one chain having the minimum edit distance. The output of Figure 5 will be processed by minimum edit distance as in Figure 6.

- Selection by probability

As it was seen in the previous example, there is maybe more than one chain with minimum edit distance. Therefore another approach is used for selecting one of these chains. The probability of the sequence word is adopted to reduce the number of chains and to select one chain. The used role, to calculate the probability of a sequence:

$$P(W_1^n) \approx \prod_{k=1}^{n} P(W_K|W_{k-1}) \quad (1)$$

Where $P(W_1^n)$ is the probability of the sequence w1…wn and $P(W_K|W_{k-1})$ represent the probability of word Wk given the preceding word Wk-1.

Figure 5. Calculating the minimum edit distance for chains

3.2.5. Reinserting of symbols

After completing the correction process for all the words in the document, the symbols, punctuations marks, and numbers that extracted before the correction process should be returned to reach a result matching the original image. These elements are re-entered into the text based on their coordinates, which include the line index and the word index.

4. RESULTS AND DISCUSSIONS

The proposed AOCR post-processing system was implemented using python 3.6 because it has many packages suitable for this task, characterized as an open-source language, can use for any platform operating system and performing various scientific calculations [26]. It was tested using the results of two commercial OCR applications, i2OCR that evaluated by (S, Vijayarani A, Sakila) as better performance among seven other systems [27] and ABBYY FINE READER that provides recognizes text quickly and accurately [28], on several images of Islamic book pages that had errors due to the poor quality of the images. Five tests of different resolutions, for 10 documents images, were made as input to each commercial application and then the suggested method is applied on the outputs. Figures 7 & 8 show graphic representation for the accuracies before and after applying the suggested method on the outputs of i2OCR and ABBYY FINE READER respectively for different image resolutions.

The accuracy of the i2OCR results is deteriorated whenever the document image spatial resolution is decreased as shown in Figure 7. The average accuracy rate system output is (96.26) at a resolution (1900 X 2687). It continues to decline until the access the average accuracy (80.51) at a resolution (1500 x 2121). The output errors are detected and corrected by the proposed post-processing system to raising the system's average accuracy to (99.55) at a resolution (1900 X 2687) until the average accuracy (92.15) at resolution (1500 x 2121).

As shown in Figure 8, the average accuracy of the ABBYY OCR system outputs is (97.52) at the spatial resolution (1240 x 1745), and it continues to go down until it reaches (75.69) at the spatial resolution (600 x 848). By the applied the proposed post-processing, the accuracy of the results is improved to be (99.49) at the spatial resolution (1240 x 1745) and (86.33) as average accuracy at the spatial resolution (600 x 848). According to results, the proposed system gives good improvement results as a postprocessing part when applied to the outputs of Arabic OCR systems. Every five words of the outputs of the Arabic OCR systems are treated as one patch in the proposed system. Table 2 shows samples of the corrected errors for the output of the used commercial systems.

After analyzing the error words of the output of the commercial systems, we see that must of them (63%) can be solved by single substitution while 37% of the errors can be solved by pair substitution, insertion, deletion or others.

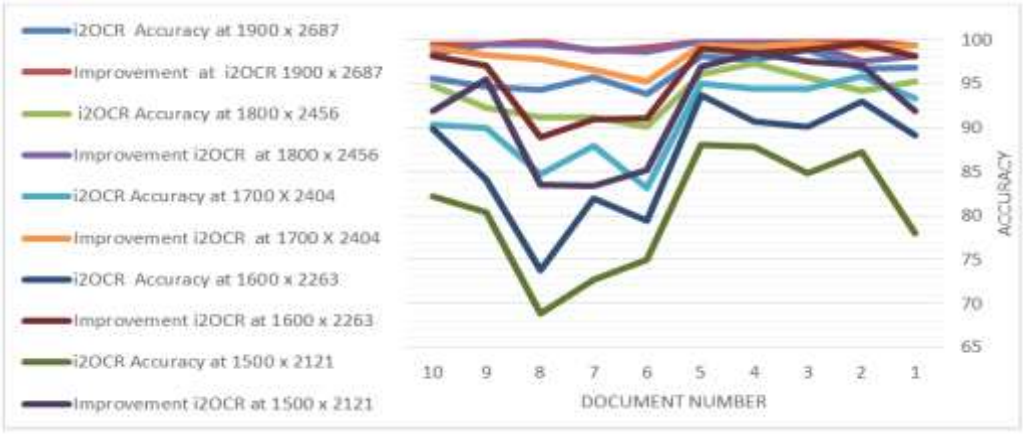

Figure 6. i2OCR improvment for resolutions 1900 X 2687, 1800 X 2456, 1700 X 2404, 1600 x 2263 and 1500 x 2121

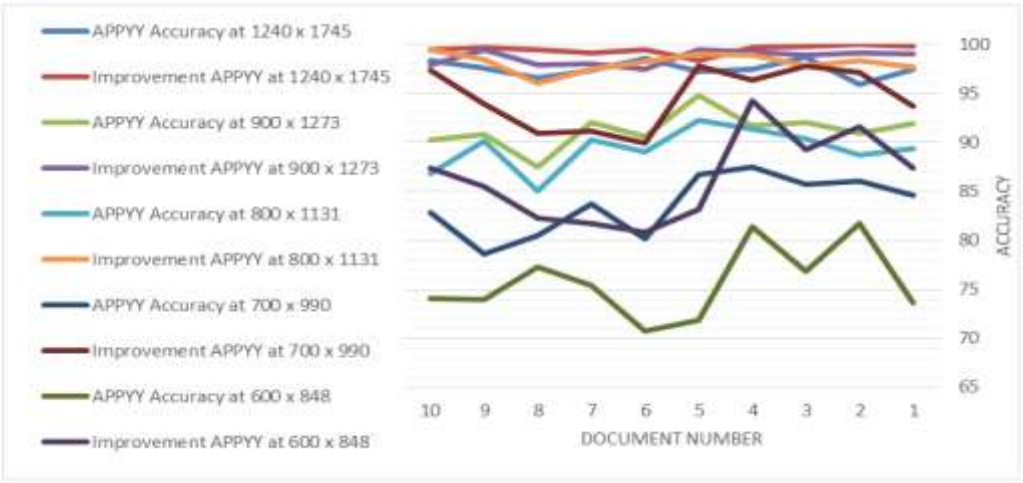

Figure 7. ABBYY OCR improvment for resolutions 1240 x 1745,900 x 1273, 800 x 1131, 700 x 990 and 600 x 848

Table 2. Samples of the corrected errors

Se	Error Phrase	Corrected Phrase
1	"اقسبل لك إن طلس الحقيتة"	"اقول لك إن طلب الحقيقة"
2	"والوصسول إليبا هو مرادناا ومبتعانا"	"والوصول إليها هو مرادنا ومبتغانا"
3	"اعن طريق النليل الفلمسفي العقلير"	"عن طريق الدليل الفلسفي العقلي"
4	"وأجلي عن بصسرك الظلمة الت"	"وأجلي عن بصرك الظلمة التي"
5	"الخبر الصسحيح الذي رواهن الشيخ"	"الخبر الصحيح الذي رواه الشيخ"

5. CONCLUSIONS AND FUTURE WORKS

In this paper, Corpus-based error correction was introduced for correcting the output of AOCR system. The proposed system is based on dictionary and N-gram language model LM constructed from the huge corpus. The experiments showed, as can be seen from the results, very good improvement in correction of errors of AOCR systems for both types real-word error and non-word errors. In other words, the system relied on the context of the word in error correction as well as the validity of word. As we can see from the results, the system still can correct errors in spite of the drop down of accuracy of the commercial systems result from the image resolution. The average of correction approximately is 7.96 % where there is a case the correction is 15.35%. The proposed system can be improved by using huge balanced corpus that covers all domains. Also, the system can work on web pages after using the PostgreSQL database engine, which works on a client-server model, for good performance.

REFERENCES

[1] A. F. H. Alharan, H. K. Fatlawi, and N. S. Ali, "A cluster-based feature selection method for image texture classification," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 14, no. 3, pp. 1433-1442, 2019, doi: 10.11591/ijeecs.v14.i3.pp1433-1442.

[2] H. K. Fatlawi, A. F. H. Alharan, and N. S. Ali, "An efficient hybrid model for reliable classification of high dimensional data using k-means clustering and bagging ensemble classifier," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 24, pp. 8379-8398, 2018.

[3] Y. Ouadid, A. Elbalaoui, M. Boutaounte, M. Fakir, and B. Minaoui, "Handwritten tifinagh character recognition using simple geometric shapes and graphs," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 13, no. 2, pp. 598-605, 2019, doi: 10.11591/ijeecs.v13.i2.pp598-605.

[4] W. Magdy and K. Darwish, "Arabic OCR error correction using character segment correction, language modeling, and shallow morphology," *COLING/ACL 2006 - EMNLP 2006 2006 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, no. July, pp. 408-414, 2006, doi: 10.3115/1610075.1610132.

[5] Z. Q. Al-Zaydi and H. Salam, "Multiple Outputs Techniques Evaluation for Arabic Character Recognition," *Int. J. Comput. Tech.*, vol. 2, no. 5, pp. 2-8, 2015.

[6] K. H. Abdulkareem *et al.*, "A Review of Fog Computing and Machine Learning : Concepts, Applications, Challenges, and Open Issues," *IEEE Access*, vol. 7, no. April 2020, pp. 153123-153140, 2019, doi: 10.1109/ACCESS.2019.2947542.

[7] Y. Bassil and M. Alwani, "OCR Post-Processing Error Correction Algorithm using Google Online Spelling Suggestion," vol. 3, no. 1, 2012.

[8] S. Naz, N. H. Khan, S. Zahoor, and M. I. Razzak, "Deep OCR for Arabic script-based language like Pastho," *Expert Syst.*, no. March, pp. 1-11, 2020, doi: 10.1111/exsy.12565.

[9] J. Outifa, S. L. Aouragh, and S. El Alaoui Ouatik, "Integration of data sources in an automatic corrector of Arabic texts," *Colloq. Inf. Sci. Technol. Cist*, vol. 0, pp. 344-348, 2016, doi: 10.1109/CIST.2016.7805068.

[10] N. Sankaran and C. V Jawahar, "Error detection in highly inflectional languages," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, pp. 1135-1139, 2013, doi: 10.1109/ICDAR.2013.230.

[11] A. M. Azmi, M. N. Almutery, and H. A. Aboalsamh, "Real-Word Errors in Arabic Texts: A Better Algorithm for Detection and Correction," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 8, pp. 1308-1320, 2019, doi: 10.1109/TASLP.2019.2918404.

[12] T. Kanungo, G. A. Marton, and O. Bulbul, "OmniPage vs. Sakhr: paired model evaluation of two Arabic OCR products," *Doc. Recognit. Retr. VI*, vol. 3651, no. January, p. 109, 1999, doi: 10.1117/12.335808.

[13] S. Verberne, "Context-sensitive spell checking based on word trigram probabilities," *Read. Writ.*, vol. 131, no. 5, pp. 3-509, 2002, doi: 10.1007/s11145-006-9040-z.

[14] E. Sciences and T. Naseem, "A Hybrid Approach for Urdu Spell Checking," *Science (80-.).*, no. November, 2004.

[15] K. Shaalan, Y. Samih, M. Attia, P. Pecina, and J. Van Genabith, "Arabic word generation and modelling for spell checking," *Proc. 8th Int. Conf. Lang. Resour. Eval. Lr. 2012*, pp. 719-725, 2012.

[16] I. A. Doush and A. M. Al-Trad, "Improving post-processing optical character recognition documents with Arabic language using spelling error detection and correction," *Int. J. Reason. Intell. Syst.*, vol. 8, no. 3-4, pp. 91-103, 2016, doi: 10.1504/IJRIS.2016.10003960.

[17] I. Q. Habeeb, "Hybrid Model of Post-Processing Techniques for Arabic Optical Character Recognition Doctor of Philosophy," 2016.

[18] A. H. Gharibeh, "A Hybrid Approach for Arabic OCR Post-Processing Using Rule Based and Word Context Techniques," *J. Chem. Inf. Model.*, vol. 53, no. 9, p. 287, 2016, doi: 10.1017/CBO9781107415324.004.

[19] S. Yousfi, S. A. Berrani, and C. Garcia, "Contribution of recurrent connectionist language models in improving LSTM-based Arabic text recognition in videos," *Pattern Recognit.*, vol. 64, no. November 2016, pp. 245-254, 2017, doi: 10.1016/j.patcog.2016.11.011.

[20] I. A. Doush, F. Alkhateeb, and A. Hamdi, "A novel Arabic OCR post-processing using rule-based and word context techniques," *Int. J. Doc. Anal. Recognit.*, 2018, doi: 10.1007/s10032-018-0297-y.

[21] T. Alvarez-l and M. Fern, "A Proposal for Book Oriented Aspect Based Sentiment Analysis", *Springer International Publishing*, vol. 2, 2018.

[22] A. Chouigui, O. Ben Khiroun, and B. Elayeb, "*Related terms extraction from Arabic news corpus using word embedding*", *Springer International Publishing*, vol. 11231 LNCS., 2019.

[23] A. Chouigui, O. Ben Khiroun, and B. Elayeb, "ANT corpus: An Arabic news text collection for textual classification," *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2017-Octob, pp. 135-142, 2018, doi: 10.1109/AICCSA.2017.22.

[24] N. Y. Habash, "Introduction to Arabic natural language processing", vol. 3, no. 1. 2010.

[25] M. Attia, P. Pecina, Y. Samih, K. Shaalan, and J. Van, "Arabic spelling error detection and correction," 2015, doi: 10.1017/S1351324915000030.

[26] A. Kumar and S. P. Panda, "A Survey: How Python Pitches in IT-World," *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 248-251, 2019, doi: 10.1109/COMITCon.2019.8862251.

[27] V. S and S. A, "Performance Comparison of OCR Tools," *Int. J. UbiComp*, vol. 6, no. 3, pp. 19-30, 2015, doi: 10.5121/iju.2015.6303.

[28] S. J. Jang, "Ocr related technology trends," vol. 8, no. 1, pp. 13-20, 2020.