Image and Text Correction Using Language Models

Ido Kissos
School of Computer Science, Tel Aviv University
Ramat Aviv, Israel

Nachum Dershowitz
School of Computer Science, Tel Aviv University
Ramat Aviv, Israel

***Abstract*—We report on experiments with the use of learned classifiers for improving OCR accuracy and generating word-level correction candidates. The method involves the simultaneous application of several image and text correction models, followed by a performance evaluation that enables the selection of the most efficient image-processing model for each image document and the most likely corrections for each word. It relies on a ground-truth corpus, comprising image documents and their transcription, plus an in-domain corpus used to build the language model. It is applicable to any language with simple segmentation rules, and performs well on morphologically-rich languages. Experiments with an Arabic newspaper corpus showed a 50% reduction in word error rate, with per-document image enhancement a major contributor.**

I. Introduction

Low quality printing, poor scanning, and physical deterioration, reduce the usefullness of many modern publicly-available digital documentary resources (books, journals, newspaper articles, etc.). Institutions are converting document images into machine readable text via Optical Character Recognition (OCR), enabling a realistic way of exploring vast document corpora with automated tools, such as indexing for textual search and machine translation. But when the images are of poor quality, the OCR task becomes notoriously difficult. Consequently, it is impossible to directly employ the obtained results for subsequent tasks, like text retrieval, without costly manual editing. For example, although contemporary OCR engines claim 97% word accuracy for Arabic, the same datasets with low-resolution images or infrequent character classes can drop below 80%.

Our proposed OCR-correction technique consists of an image pre-processing and text post-correction pipeline, based on a composite machine-learning classification. The technique wraps the core OCR engine and is in practice agnostic to it.

Image correction applies a small set of image enhancement algorithms on copies of the document images, which serve as input for the OCR engine. The enhancements include image scaling, binarization methods and parameter thresholds, and were chosen for their experimental accuracy gain. The potential gain for every algorithm is evaluated as the sum of the positive accuracy improvements, relying on a learned classifier for the selection of improved OCR text over a baseline OCR text, that is the output of the image with the OCR's engine's default pre-processing. Such a classifier was trained with a ground-truth set, relying on recognition confidence statistics and language model features to output an accuracy prediction.

Text post-correction applies a lexical spellchecker, and potentially corrects single-error misspellings and a certain class of double-error misspellings, which are the major source of inaccurate recognition in most OCR use-cases. It takes into consideration several valuable word features, each giving additional information for a possible spelling correction. It comprises two consecutive stages: (a) word expansion based on a confusion matrix, and (b) word selection by a regression model based on word features. The confusion matrix and regression model are built from a transcribed set of images, while the word features rely on a language model built from a large textual dataset. The first stage generates correction candidates, ensuring high recall for a given word, while the second assures word-level precision by selecting the most probable word for a given position. Relying on features extracted from pre-existing knowledge, such as unigram and bigram document frequencies extracted from electronic dictionaries, as well as OCR metrics, such as recognition confidence and confusion matrix, we accomplished a significant improvement of text accuracy.

The two correction methods—image enhancement and text correction—implement equivalent methodologies: both begin by promoting recall, generating many correction candidates that some may improve the baseline result, and afterwards use prior knowledge and context to gain precision, namely selecting the best candidate. Initially, our research focused on the text method; it is its success that pushed the idea on us of implementing a similar methodology for image processing, which in the end turned out to give the larger gain in accuracy.

We report on experiments with applying the methodology to Arabic, using test data from the "Arabic Press Archive" of the Moshe Dayan Center at Tel Aviv University. There are a number of open-source and commercial OCR systems trained for Arabic [1]; we used NovoDynamics NovoVerus commercial version 4, one of the leading OCR engines for Arabic scripts. For evaluation purposes we use the Word Error Rate (WER) measure, or its complement with 1, named *OCR accuracy*, which is adapted for subsequent applications of the OCR output, such as information retrieval. Our correction method performs effectively, reducing faulty words by a rate of 50% on this dataset, which is an 8% absolute improvement in accuracy. The overall results showed negligible false-positive errors, namely the method rarely rejects correct OCR words in favor of an erroneous correction, which is a major concern in spellcheckers. An analysis of classifier performance shows that bigram features have the highest impact on its accuracy,

suggesting that the method is mainly context reliant.

Section II presents the image enhancement methodology and Section III, the text correction methodology. The main section, Section IV, provides and discusses the experimental results. It is followed by a brief discussion.

II. Image Enhancement

The ability to consistently select the best image processing for every image document leans on the capability to reliably predict its performance, namely, its OCR text accuracy. To facilitate this task, this prediction can be based on the extracted text of each image and not on the image itself, suggesting that an a posteriori selection of image processing algorithm could outperform the common a priori one.

The enhancement method requires one to move from a single-pass OCR engine, in which every document is processed once—and for which OCR engines are optimized, to multi-pass OCR. The latter enables an accuracy-performance trade-off, promoting better OCR results at the compromise of CPU resources, which is often a reasonable call for digitization projects. Having several output texts for a single image document, we can rank them and choose the most accurate, according to our prediction, for a specific image.

The multi-pass architecture is built as a pipeline where each module applies a family of dependent algorithms, for example binarization methods and thresholds, and the sequential modules are independent one of the other. After every module an evaluation sequence extracts the document image text and predicts its accuracy, then feeds its processed image to the next module. This implementation avoids the application of an unfeasible number of image processing sets, which is the sum of all possible algorithm combinations. Assuming independence between the modules, their application order has only a small significance.

Each module comprises two stages: (1) Enhancement candidate generation – Every algorithm in the set renders a processed image that serves as an input to the OCR engine. (2) Candidate evaluation – For each candidate, language model features and confidence statistics are extracted from its OCR text output. These are used to rank the candidates according to their likely correctness, while the highest ranked candidate is selected as the image for the subsequent module, or the text for post-correction if it is the last module.

Candidates are generated based on a set of image processing algorithms and thresholds we found had positive effect on the corpus' OCR accuracy. Finding this set required an evaluation of the potential gain of every algorithm. We benchmarked the performances of a large set of algorithms that are commonly used for OCR pre-processing, tuned their parameters, and chose several configurations for each algorithm type i that produced the highest gains, which we denote by set X_i. In order to comply with the metric of improving average accuracy, finding X required the solution of an optimization problem that can be formulated as follows:

$$\arg\max_{X_i} \sum_{doc \in \textit{training-set}} accuracy(doc|X_i)$$

subject to

$$X_i \subseteq \{\text{Scale, Binarize, Denoise}\} \times \textit{Algorithm Configurations}, |X_i| \leq 3$$

The limitation to a set of size 3 is conditioned for calculation purposes and empirical gain bounds. The approximation for X_i is obtained by trial and error and stopped when reaching negligible accuracy improvements.

Each family of dependent methods or thresholds is implemented in a separate module, resulting in a total of three modules. The following algorithm types and thresholds were applied: (a) Bicubic and K-Nearest-Neighbors methods, scaling from 1 to 3 with 0.25 stepsize. (b) Sauvola and threshold-based binarization algorithms, with thresholds varying from 100 to 250 with a step size of 25. (c) Image denoising methods included three different filters: Mild, Median and None, enabled in NovoVerus. Based on the above close-to-optimal algorithm set, the candidate generation module applies this set to the input image and extracts its text for the evaluation phase.

The evaluation stage evaluates the textual output from each of the image candidates of every module. It is based on a learned linear regression that ranks the candidates according to their expected accuracy. This score does not necessarily have to be a normalized accuracy, but as a language model score, assessing which textual output is more probable to occur. As for a typical machine learning algorithm, we extract features and train a regression upon them. The regression relies on the language features of bigram occurences, as well as a confidence metric. The latter is calculated based on the character level confidence given by the OCR engine, aggregated to a document level statistic by averaging over words, namely $\sigma_{doc} = \sum_{word \in doc} \frac{\min_{char \in word} \textit{conf}_{char}}{|doc|_w}$.

We train the accuracy prediction regression on the labeled set for which we know the real accuracy, and try different feature representation and models to achieve good results. Every image is scored independently from all other images, including its enhancement candidates. The loss-function the regression model is assessed upon the accuracy loss of faulty ranking, that can be formulated as follows:

$$\arg\min_{\hat{f}} L(f, \hat{f}) = f - \hat{f}$$

where

$$f_{doc} = \max_{\textit{image-candidate}} accuracy(doc|\textit{image-candidate})$$

$$\hat{f}_{doc} = accuracy(doc)$$

III. Text Correction

The OCR error model is vital in suggesting and evaluating candidates. At the heart of the error model is a candidate generation for correction, based on a confusion matrix giving conditional probabilities of character segment edits. The possible error corrections include the primitive 1 Levenshtein

```
<wrongSegment string="ر">
  <correctSegment popularity="14">ز</correctSegment>
  <correctSegment popularity="11">ى</correctSegment>
  <correctSegment popularity="10">ب</correctSegment>
  <correctSegment popularity="2">و</correctSegment>
  <correctSegment popularity="2">م</correctSegment>
  <correctSegment popularity="1">ة</correctSegment>
  <correctSegment popularity="1">ل</correctSegment>
  <correctSegment popularity="1">ا</correctSegment>
</wrongSegment>
```

Fig. 1. Excerpt of the confusion matrix for the character *raa*.

edit distance,[1] as well as spacing (word segmentation) errors. We focus the discussion on the word level at a certain position in the text, which is obtained by a standard tokenization of the OCR output text.

The error correction methodology comprises three stages: (1) Correction candidate generation – The original word is expanded by a confusion matrix and a dictionary lookup, forming all together a correction-candidates vector. (2) Candidate evaluation – Based on the candidate extracted features extracted, it is scored according to their correctness probability at this position and ranked among the other candidates. (3) Word classification – Selects the most probable word between the original word and the highest-ranked correction candidate.

Correction candidates are generated based on the observed OCR error model, represented by a weighted confusion matrix. This model was built by aligning the ground truth image document to its respective OCR text at word level. For example, Figure 1 shows an excerpt of this representation, that is, as expected, very affected by the characters' graphical resemblance. The candidate generation is rule-based, where every character segment in a word is looked up in the confusion matrix and replaced by a possible segment correction.

The candidate evaluation stage produces an ordered word vector of correction candidates. This stage does not take into account the original OCR output, as it has different features and will be considered in a secondary stage. As a preliminary stage, the input vector is cleaned from all non-dictionary words. As the dictionary is based on a large corpus, this procedure has only a negligible deleterious effect, while discarding a considerable number of irrelevant candidates, hence facilitating scoring. In a secondary stage the word score is calculated by a trained regression model using the word's features as input: (a) Confusion weight – The weight attribute of the corruption-correction pair in the confusion matrix, which is the number of occurrences of this pair calculated by the noisy channel on the training set. (b) Unigram frequency – The unigram document frequency, providing a thematic domain and language feature independent of adjacent words or document context. (c) Backward/Forward bigram frequency – The maximal document frequency of the bigram formed by a correction candidate and any candidate at the preceding/following position. This feature is valuable as it contains an intersection between language model and domain context, but is non-existent for many of the bigrams. No subsequent normalization procedure had to be made in order to linearize the feature effect for later linear regression modeling. In other words, the confusion weight behaves linearly, as well as the term frequency features that proportionally promote frequent corrections relative to their appearance in a similar corpus.

The regression was trained from the OCR erroneous word set, comprised of words the candidate generator supposedly generates. We used the training words to generate their correction-candidates vector with their extracted features, with the single correct candidate marked with a positive output. Appending these vectors creates a large training set used to create a regression model that attributes a continuous score to every candidate. The model is used to rank correction-candidate vectors and to sort them in descending order.

Subsequently, we train a classifier to decide whether the OCR word should be replaced with its highest ranked correction-candidate. Such a replacement is made in case the candidate is more likely to be the correct word at this position. We will refer to the OCR word and its highest-ranked correction-candidate as an "correction pair". The detection of an erroneous OCR word in the case of real-word errors, also referred as false friend, is a difficult task. Such cases are frequent in Arabic due to its morphological richness. Prior work in [2] suggests a shallow language model to handle these.

Our classifier relies on the OCR confidence of the word, as well as the correction pair's proportional language features present in the previous classifier, for example the forward-bigram proportion. The proportion representation forms a comparative features that has a linear sense. A simple smoothing method was used to handle null-occurrences.

The correction decision is made by a model trained on the total corpus, except words that do not have a their correction generated. Pairs with erroneous OCR word and correct candidate are marked with a positive output, indicating that these cases are suitable for replacement.

IV. Testing the model

The model was trained on 211 image articles, scanned from *Al-Hayat* newspaper from 1994. The set includes the articles' ground truth transcription and the OCR outputs from Novovo-dynamic's NovoVerus. Another 50 labeled articles were left aside as test set for evaluation,[2] adding up to a total of 22,000 words. The language model was trained with the large in-domain corpus of Arabic Gigaword.

A. Image Enhancement

The non-sequential accuracy gains for image enhancement are summarized by algorithm type in Table I, summing up to a total of 5.7% absolute average accuracy improvement when algorithms are applied sequentially. The accuracy changes caused by different algorithms have a large variance, induced by several articles with a considerable improvement, while

[1]Based on a modified Levenshtein distance, where further primitive edit operations (character merge and split) are used (also known as 2:1 and 1:2 alignments).

[2]The entire dataset is publicly available for research purposes at https://github.com/idoki/ocr_correction.

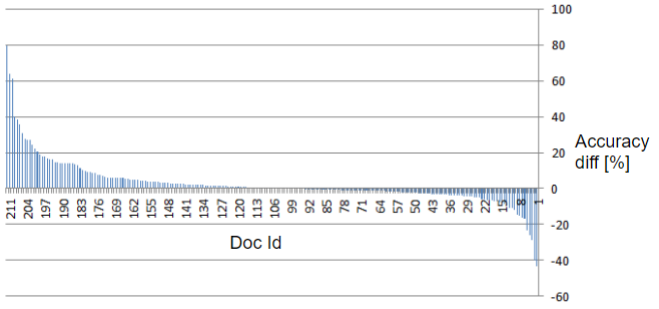

Fig. 2. Accuracy difference ordered per document. Bi-cubical Scale; Threshold = 1.5. As a single algorithm, it had the greatest average positive gain (4.4% in accuracy), namely almost 30% WER decrease.

the rest of the corpus' accuracy remains unaffected or decreases. This phenomenon can be seen by Figure 2. The major achievement of the method is its ability to exploit the potential gain of each algorithm, without suffering much from any potential loss. This can be attributed to the regression function that efficiently predicts the relative accuracy of the different candidates, as demonstrated by its loss-function performance in Table II. This overwhelming results can be attributed to the relative easiness of the task, as it relies solely on the differences between the images induced by different algorithms, which generally affect only a small subset of words that would produce better features for better extractions.

TABLE I
ALGORITHM SET SELECTION WITH AVERAGE ACCURACY GAIN

Module	Algorithm [:Threshold]	Avg. Gain
Scale	Default:1, Bicubical:1.5, Bicubical:2.25	5.3%
Binarize	Default, Sauvola:170, Sauvola:230	1.6%
Denoise	Default:none, Mild, Median	0.4%

TABLE II
PERFORMANCE OF THE IMAGE CANDIDATE CLASSIFIER

	Selection of best candidate
0-1 Loss	4/50
Avg. Weighted Loss	0.4%

B. Text Correction

This text-correction phase assumes the OCR text of the optimal image as input, not implying any obligation for the first stage to occur for the lexical correction, but rather to present the results in a standardized way.

The method is a pipe of 3 subsequent modules that can be seen as a funnel that narrows down from the correction-candidate generation through the ranking of these words to the classifier that decides of the replacement of the original word with the highest ranked correction candidate. Therefore, these stages are evaluated independently, allowing a ceiling analysis for ongoing improvements.

An analysis of the error type distribution on the test set demonstrates that 60% of the erroneous words had been

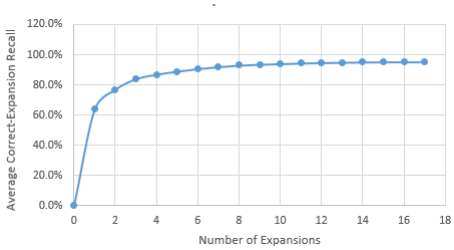

Fig. 3. Average recall on erroneous words as function of correction-candidates.

retrieved in their correct spelling in the correction-candidate generation process. The non-corrected words either did not belong to primitive 1-Levenshtein misspellings, or their correction instance did not occur in the training set. This fair result suggests that the OCR errors belong to a wider error set than the one trained on and can be attributed to random text variability, such as noise or deterioration, or to the existance of low graphical resemblance between large sets of characters. This result sets an upper bound to the correction efficiency, that would be reached only if the subsequent correction tasks, namely ranking and correction decision, are fully efficient. Improving it could be acquired by enlarging the training set or by generating candidates by additional logic.

The score for each candidate is attributed by a trained a logistic regression, yielding the results shown in Figure 3. Calculated for words that have a valid candidate, the best model is able to find the proper correction within the top 5 proposed candidates for 90% of the words, and within the highest ranked candidate for 64% of the words. Improving this result may be achieved by a better model, such as a non-linear one, or by expanding the training set in order to enhance the confusion weight feature. Another way to overcome this caveat is taking into account more than the top candidate and canceling the next phase. The text output would contain multiple words on the same position, complying with the goal of improving retrievabiliy on image documents.

Table III reports the decision model performance over all words in text. The critical factor in this stage is the false positive rate, namely rejecting a correct OCR word in favor of its correction-candidate, as most of OCR words are correct and such rejections would significantly harm the reliability of the method. Therefore, the trained model gives preference to false positive rate diminution over false negative diminution. The main reason for this good result, implying an efficient classification model, is the bigram proportion feature. In case left or right bigram exists, as occurs in vast majority of cases on correct words thanks to the large corpus on which is based our language model, the respective feature has a high impact on the classifier and would generally lead to a righteous correction.

C. Overall Results

An overall representation of the results over the test set is shown in Figure 4. The baseline OCR text WER on the test

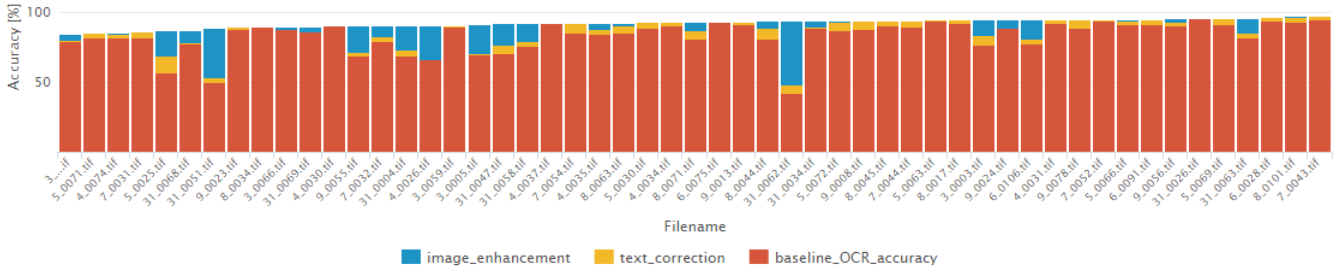

Fig. 4. Accuracy over the test set.

TABLE III
PERFORMANCE OF THE DECISION MODEL FOR WORD CORRECTION

	OCR word	
	correct	incorrect
Reject OCR word	2%	94%
Accept OCR word	98%	6%

collection is 16.5% on average; applying image enhancement reduces it to 10.4%, while applying on top of that the text correction results to a 7.9% WER, reducing the overall error by over than 50%. This is a considerable improvement given that improvement is harder as WER gets lower. This relative accuracy improvement suggests that this method belongs to the state-of-the-art algorithms for OCR correction. A further examination of the uncorrected errors demonstrates that most originate from deteriorated zones or significant inaccuracies in OCR recognition. The rigorous implementation of the image enhancement and lexical correction methods, shown in [3] and other works, such as [4], and especially their combination by machine-learning techniques, bring most of the additive improvement gains suggested in these.

Image enhancement improves almost twice as much compared to lexical correction. That can be explained by the fact that improving input is generally better than correcting the output, as information is added to the process and exploited in the subsequent OCR tasks. The overall results of the image enhancement demonstrates that the algorithms family are rather dependent by the fact that the overall accuracy gain is not very close to the addition of its two modules, 5.7% vs 7.3%. Nevertheless, as most of the gain is related to articles with an exceptional improvement, and not an average improvement, the accuracy figure should be taken with care as it is sensitive to the data.

The ceiling analysis for the lexical correction clearly designates the correction candidate generation as a weak link, due to the fact that it does not handle out of primitive 1-Levenshtein misspellings, as well as its relatively low generalization on specific error type, such as spacing errors, missing in total more than 35% of the true candidates in their generation process. Adding other correction methods to the current noisy channel one, training based as well as unsupervised methods, would greatly improve the overall process. The ranker could also be improved by working on its accuracy for the highest ranked candidates, as for 35% of erroneous words their correction is among the top 5 ranked candidates but does not make it to the top candidate, which is the only one to make it to the subsequent correction decision. The correction decision maker is effective; with its large training set and indicative features one can expect similar results for different datasets.

V. CONCLUSIONS

This work examined the use of machine-learning techniques for improving OCR accuracy by using the combination of a number of features to enhance an image for OCR and to correct misspelled OCR words. The relative independence of the features, issuing from the language model, OCR model and document context, enables a reliable spelling model that can be trained for many languages and domains. The results of the experiment on Arabic OCR text show an improvement in accuracy for every additional feature, implying the superiority of our multi-feature approach over traditional single-feature approaches that most spelling correction techniques rely on. We can infer from the bigram feature significance that the contextual word completion is a reliable method for a machine as well for the human eye. Lastly, we would like to emphasize the similarity between image enhancement and text correction. Even though both are considered unrelated domains, viz. vision and language, this work and its results demonstrate the mutual significance of the two and mostly the ability to apply a similar correction methodology to both.

The strength of this method is its ability to "squeeze out" the performance of any out-of-the-box OCR engine. Although new OCR methods based on deep learning techniques are emerging and start to commercialize, taking a step further the standard OCR engines' accuracy, these results compare to state-of-the-art techniques, while a combination of this method to these recent techniques may bring even further improvement.

REFERENCES

[1] V. Märgner and H. El Abed, Eds., *Guide to OCR for Arabic Scripts*. London: Springer, 2012.
[2] U. Reffle, A. Gotscharek, C. Ringlstetter, and K. Schulz, "Successfully detecting and correcting false friends using channel profiles," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 12, no. 3, pp. 165–174, 2009.
[3] K. Kukich, "Techniques for automatically correcting words in text," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 377–439, Dec. 1992.
[4] M. Tillenius, "Efficient generation and ranking of spelling error corrections," NADA, Report TRITA-NA-E9621, 1996.