

Automatic Assessment of OCR Quality in Historical Documents

Anshul Gupta¹, Ricardo Gutierrez-Osuna¹, Matthew Christy², Boris Capitanu³,
Loretta Auvil³, Liz Grumbach², Richard Furuta¹, and Laura Mandell²

¹Department of Computer Science and Engineering, Texas A&M University

²Initiative for Digital Humanities, Media, and Culture, Texas A&M University

³Illinois Informatics Institute, University of Illinois at Urbana-Champaign

{anshulg,rgutier,mchristy,egrumbac,furuta,mandell}@tamu.edu, {capitanu,lauvil}@illinois.edu

Abstract

Mass digitization of historical documents is a challenging problem for optical character recognition (OCR) tools. Issues include noisy backgrounds and faded text due to aging, border/marginal noise, bleed-through, skewing, warping, as well as irregular fonts and page layouts. As a result, OCR tools often produce a large number of spurious bounding boxes (BBs) in addition to those that correspond to words in the document. This paper presents an iterative classification algorithm to automatically label BBs (i.e., as text or noise) based on their spatial distribution and geometry. The approach uses a rule-based classifier to generate initial text/noise labels for each BB, followed by an iterative classifier that refines the initial labels by incorporating local information to each BB, its spatial location, shape and size. When evaluated on a dataset containing over 72,000 manually-labeled BBs from 159 historical documents, the algorithm can classify BBs with 0.95 precision and 0.96 recall. Further evaluation on a collection of 6,775 documents with ground-truth transcriptions shows that the algorithm can also be used to predict document quality (0.7 correlation) and improve OCR transcriptions in 85% of the cases.

Introduction

Optical character recognition (OCR) of historical texts in the hand-press period (roughly 1475-1800) is a challenging task due to the characteristics of the physical documents and the quality of their scanned images. Early printing processes (printing presses, mass paper production, hand-made typefaces) produced texts with fluctuating baselines, mixed fonts, and varied concentrations of ink, among many other irregularities. To make matters worse, the existing digital collections for documents of that period largely consist of binary (i.e., as opposed to grayscale), low-quality and low-resolution images, the result of digitization from microfilm converted from photographs –four decades

and three generations away from the originals.

Motivated by these issues, in 2013 we started the Early Modern OCR Project (eMOP; <http://emop.tamu.edu>) with funding from the Andrew W. Mellon Foundation. eMOP is a two-year mass digitization project that seeks to improve OCR for some 45 million pages from the Eighteenth Century Collections Online (ECCO) and Early English Books Online (EEBO) proprietary database products. Our goal extends beyond producing accurate transcriptions for these collections, and also aims to create tools (dictionaries, workflows, and databases) to support scholarly research at libraries and museums. Much like our team, these organizations lack the resources to manually transcribe their collections or contract with commercial OCR services (e.g., Prime Recognition Corp.) As such, and as required by our sponsor, all tools used and produced by eMOP must remain free or open-source.

As a step towards this goal, this paper describes an approach to assess the quality of historical documents that does not require image processing or human tagging. As illustrated in Fig. 1, when a document has poor quality, the OCR engine generally produces a large number of spurious bounding boxes (BBs) in addition to those that correspond to words in the document. As we will show, it is possible to discriminate between noisy and text BBs by analyzing statistical differences in their shape, size, position and confidence score returned by the OCR engine. This approach is advantageous for two main reasons. First, it does not require dedicated image processing algorithms (Farahmand, Sarrafzadeh et al. 2013), which can become prohibitive for large document collections. Second, the approach is language-agnostic because it relies exclusively on geometrical properties of BBs rather than the text transcription associated with them.

In the sections that follow, we propose an iterative relabeling algorithm to classify BBs into text or noise, and validate it on a dataset containing 159 mid-to-poor quality documents (over 72,000 manually-labeled BBs). Then, we illustrate how the algorithm can be used to obtain an

objective measure of document quality and improve OCR transcription performance by filtering out noise BBs before the document undergoes post-correction.

Background & Related Work

The ability to triage documents is critical in large-scale document digitization. Document triage prevents heavily-degraded documents from entering the OCR pipeline, and instead directs them elsewhere for additional processing (e.g., rescanning, image denoising). In these cases, quality is generally defined as an objective property of the document image, such as OCR accuracy, though subjective measures (e.g., Mean Opinion Scores) have also been used. Image features that have been found to correlate with OCR performance include global properties, such as the amount of black background speckle, image sharpness and uniformity, as well as local properties of the text, such as stroke thickness and continuity, and character/word height-to-width ratio (Ye and Doermann 2013).

A few studies have focused on improving OCR performance by pre-applying image restoration techniques, such as deblurring, skew removal, and bleed-through removal, to mention a few. As pointed out by Lins et al. (2010), however, these techniques should not be blindly applied but should be used selectively based on the type of noise or degradation present in the document. For this purpose, the authors developed a method to identify five types of noise (bleed through, skew, orientation, blur and framing) based on image features such as palette, gamut, or number of foreground pixels. The authors found that the overhead of this noise-classifier was far lower than running the image through unnecessary filters. In related work, Sandhya et al. (2012) developed a taxonomy of image noises in historical documents that extends beyond the five categories of Lins et al. (2010). Their taxonomy considered four types of noise sources: aging, digitization and storage, physical factors (e.g., folding, burn, bleed-through) and document factors (e.g., varying fonts, mixed alphabets.) More recently, Farahmand et al. (2013) reviewed image processing techniques to remove ruled-line noise, marginal noise, clutter noise, stroke-line pattern noise, background noise, and salt-pepper noise.

A number of studies have focused on post-correcting errors in OCR outputs by modeling typographical variations in historical documents; see (Reynaert 2008; Reffle and Ringlsetter 2013) and references therein. As an example, Alex et al. (2012) proposed two OCR post-correction methods for the problems of end-of-line hyphen removal and substitution of *long-s* (recognized as *f*) to letter *s* (e.g. “*fenfible*” to “*sensible*”). Using dictionary-based methods, the authors reported a 12.5% reduction in word error rates. For these techniques to be effective, however, noise BBs must be removed in advance.

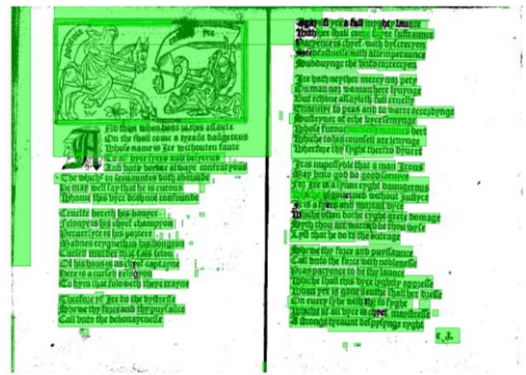


Fig. 1. OCR output for an eMOP document; BBs shown in green

Methods

Our pipeline is based on the Tesseract open-source OCR engine available from Google (Smith 2007). For each document image, Tesseract produces an hOCR data file (an open standard for formatted text from OCR) containing the layout and logical structure of the document, including the coordinates of the BB for each recognized word along with its text transcription and recognition confidence. It is the hOCR file, not the underlying image, that we use for analysis. Our overall approach for discriminating text and noise BBs is illustrated in Fig. 2. The individual steps (pre-filtering, column segmentation, and local iterative relabeling) are described next.

Pre-Filtering

The first step in the process consists of generating initial labels for each of the BBs returned by Tesseract. For this purpose we use a rule-based classifier that considers three features for each BB: word confidence, height-to-width ratio and area. The rules are derived as follows:

- **Rule 1: OCR word confidence.** BBs with very low or very high confidence predominantly consist of noise, and are flagged accordingly during pre-filtering.
- **Rule 2: Height-to-width ratio.** Most words are written horizontally, so the height-to-width ratio is generally lower for word BBs than for noise BBs. Consequently, if this ratio is less than a threshold we label the BB as text; otherwise, we label it as noise.
- **Rule 3: Area.** Tesseract has a tendency to misidentify speckles as legitimate text; fortunately, these areas are small as compared to normal text BBs. Accordingly, we label as noise all BBs in the lowest percentiles of the total area for the document.

Thresholds for the individual rules are optimized simultaneously with a manually-labeled subset of the corpus; see results section. The final filter is the conjunction of the three rules. BBs classified as text at this stage are used in the next stages to extract column layout and estimate the average font size of each document.

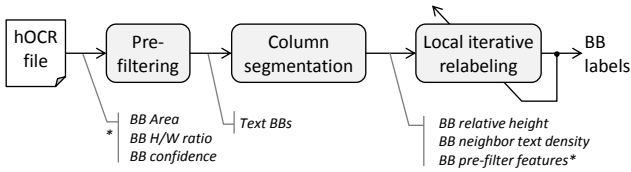


Fig. 2. Overview of the proposed BB classification method and features used for recognition at each stage

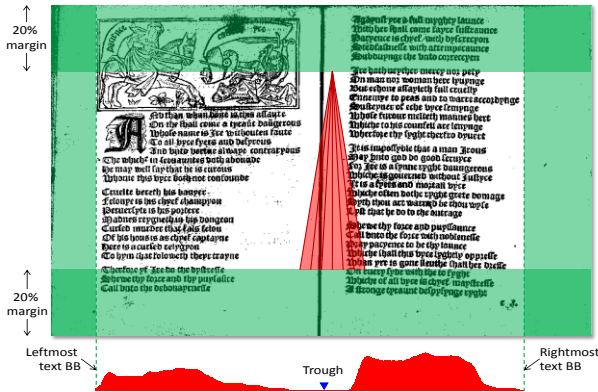


Fig. 3. Segmenting columns by identifying troughs in the horizontal distribution of BBs

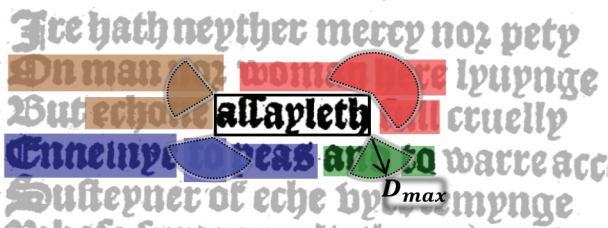


Fig. 4. Finding nearest neighbors. Only those within D_{max} from the corners of the target BB (outlined) are considered. Colors indicate the corner to which neighbors are assigned

Column segmentation

Documents in the eMOP collection generally have multiple pages and/or columns, each with its own set of problems (e.g., degree of skew or noise). For this reason, the second step in the process consists of dividing each image into its constituent pages and columns, so that each can be processed individually. First, we identify the leftmost and rightmost text BB from the pre-filtering stage; these coordinates define the text boundaries of the image. Then, we perform column segmentation by analyzing the distribution of BBs over the horizontal axis; the dominant troughs in this distribution define the column boundaries.

To compute this distribution of BBs, we divide the horizontal axis with 1000 evenly-spaced points. At each point, we trace rays from the top margin to the bottom margin with slopes in the range $90^\circ \pm 3^\circ$ in increments of 0.2° , then calculate the number of intersecting BBs for each ray. At each point, we then identify the ray with the fewest intersections, and that becomes the value of the

distribution at that point. Since images tend to have a large number of spurious BBs at the margins, any BBs in the top and bottom 20% are discarded. The overall process is illustrated in Fig. 3.

Local iterative relabeling

After each page has been split into columns, we apply an iterative relabeling algorithm to the BBs of each column. The rationale behind this final step is that BBs surrounded by text are more likely to contain text than those surrounded by noise. Accordingly, for each of the four vertices of each BB we find its nearest neighbors (see Fig. 4). Then, we calculate a weighted score, S , based on the label of each neighbor penalized by its distance:

$$S(b) = \frac{\sum_{k=1}^P w_k L_k}{\sum_{k=1}^P w_k}, \quad \text{with } w_k = \frac{1}{\text{dist}(b,k)} \quad (1)$$

where b is the index of the BB, N is the number of BBs within distance D_{max} from the vertices of b , and P is the maximum number of nearest neighbors considered ($P \leq N$). L_k is the predicted label (0: noise; 1: text) for the k -th nearest neighbor, initially taken from the pre-filtering step. As illustrated in Fig. 4, the distance D_{max} limits the search area for nearest neighbors, preventing text BBs that are far from b to be considered in the computation. The distance D_{max} is computed relative to H_{med} , the median height of text BBs found in the pre-filtering stage, plus a tolerance defined by H_{IQR} , their interquartile range; both statistics are computed for each individual column in the image:

$$D_{max} = H_{med} + \alpha \times H_{IQR} \quad (2)$$

where α defines the tolerance; the larger its value the more distant neighbors that are allowed in the computation of S of eq. (1). In our implementation, the value of α is optimized to minimize the mean-square error between S and the ground-truth label for all BBs in a training set.

The iterative process starts by initializing BB labels with those from the pre-filtering stage. From these labels, an initial score S can be computed for each BB. This score is then combined with six additional features (see Table 1), and passed as an input to a multilayer perceptron (MLP) previously trained to classify BBs as either text or noise. The additional features include those used in pre-filtering ($C_{OCR}, H/W, A$) as well as the BB position relative to the document margins, and its height normalized to H_{med} and

Table 1. Features used during local iterative relabeling

Features	Description
S	Score from nearest neighbors ; see eq. (1)
C_{OCR}	OCR word confidence*
H/W	Height-to-width ratio of BB*
A	Area of BB*
H_{norm}	Normalized height: $H_{norm} = (H - H_{med})/H_{IQR}$
H_{dist}	Horizontal distance from the middle of the page
V_{dist}	Vertical distance from the top margin

*available from the pre-filtering stage

H_{IQR} . The resulting labels are used to re-compute S and the process is repeated until convergence, i.e., labels no longer change from one iteration to the next.

Results

Datasets

To test the proposed algorithm we generated three separate datasets (see Table 2) consisting of binarized document images from the eMOP collection, carefully selected to represent the variety of documents in the corpora. This included single column, multi-page and multi-column document images, as well as images with artifacts due to ink bleed-through, multiple skew angles, warping, printed margins, printed column separators, and pictures. Each BB returned by Tesseract for each of the document images in all three datasets was then manually labelled (i.e., text/noise) to generate ground truth data, for a total of 72,366 BBs. As labeling criteria, we considered as noise any BB that spanned more than two lines of text, as well as BBs around pictures, small speckles, and printed margins. The remaining BBs were labelled as text. To guard against differences in image size, the coordinates of BBs for each document were $[0,1]$ normalized. Dataset 1 was used to optimize thresholds in the pre-filtering stage whereas dataset 2 was used to optimize parameters α and P in the local iterative relabeling stage. Dataset 3 was used to cross-validate the MLP and evaluate overall performance.

Table 2 Datasets used for training and validation purposes

Dataset	# images	% Text/Non-Text	# BBs
1	39	69/31	14,705
2	34	71/29	15,896
3	86	66/34	41,765

Pre-filtering

Fig. 5 shows the distribution of features for noise and text BBs in the documents from dataset 1. The distribution of normalized areas in Fig. 5a indicates that noise BBs tend to be smaller than text BBs, following our observations that Tesseract has a tendency to generate small spurious BBs whenever speckle noise is present in the image. Shown in Fig. 5b, the distribution of OCR word confidence values for noise BBs is multimodal, with peaks near the extrema (0,1), whereas for text BBs it is normally distributed with a peak around 65% confidence. Finally, the distribution of H/W ratios in Fig. 5c shows clear differences between the two types of BBs, with text generally having a much lower H/W ratio, as could be anticipated.

To optimize the threshold values for the three rules in the pre-filtering stage, we performed a receiver-operating-characteristic (ROC) analysis of the binary classification problem on dataset 1. Namely, we performed exhaustive

search for the word confidence (two thresholds), height-to-width ratio and area thresholds (a 4-dimensional search space) to find the operating point with maximum F1-score on the precision-recall curve. The derived rules were:

- **Rule 1:** If $0 < C_{OCR} < 0.95$, then **TEXT**
- **Rule 2:** If $H/W < 2$, then **TEXT**
- **Rule 3:** If $A > 1^{\text{st}}$ percentile, then **TEXT**

which, when used as a conjunction, yield a F1-score of 0.93 (0.94 precision; 0.91 recall). Thus, pre-filtering can identify a significant number of noisy BBs, but it also mislabels a large proportion (9%) of text BBs in the documents. This is largely due to the fact that it does not consider information local to each BB, a problem that is handled by the last step in the process: local iterative relabeling.

Column extraction

The bottom panel in Fig. 3 illustrates the horizontal distribution of BBs for one of the images in the collection. The limits for the two columns in the document are clearly indicated by troughs in the distribution. Fig. 6 shows segmentation results for two additional and more challenging documents due to noise and skew.

Local iterative relabeling

The MLP for the iterative process consisted of a hidden layer with 8 tangent-sigmoidal neurons, and 2 output neurons (i.e., one per class) with soft-max activation function to ensure MLP outputs could be interpreted as probabilities. The number of hidden units ($N_H = 8$) was optimized through three-fold cross-validation over dataset

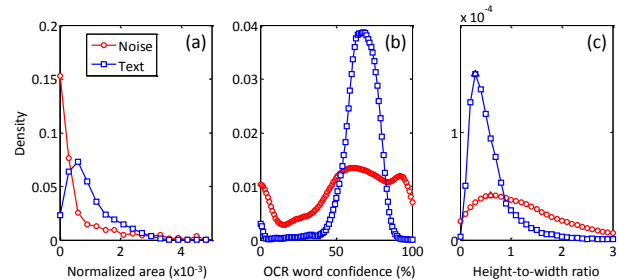


Fig. 5. Feature distributions for BBs in dataset 1

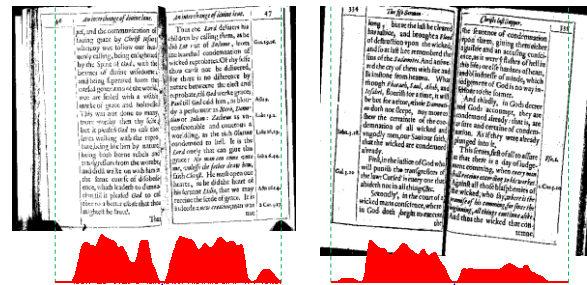


Fig. 6. Column segmentation for two difficult test cases

3 with the F1-score as the objective function. Parameter P in eq. (1), the maximum number of neighbors, was set to 84 (21 per vertex), and parameter α in eq. (2), was set to 10. These optimal values were extracted by minimizing the mean square error between S and the ground-truth label for all BBs in dataset 2.

We also performed three-fold cross-validation over dataset 3 to compare model performance before and after iterative relabeling. Results are summarized in Fig. 8a; precision, recall and the F1 score improve when compared to pre-filtering results on dataset 3, with the largest gains obtained for recall (from 0.89 to 0.96). Fig. 8b summarizes the convergence rate; in 95% of the cases the algorithm converges within three iterations.

Fig. 7 shows a document overlaid with the BBs returned by Tesseract. The fill color (green vs. red) represents the MLP prediction (text vs. noise, respectively), with higher color saturation denoting higher confidence; see color-bar insert. Arrows 1 and 2 illustrate two cases for which prediction was correct but the MLP had low confidence, hence the gray tone. Arrow 3 points to a BB that covers graphics and a decorative drop cap, neither of which is likely to lead to a good OCR transcription. Finally, arrow 4 points to a BB that contains two lines of text; as such, the OCR transcriptions are likely to contain garbage.



Fig. 7. Iterative relabeling results for the image in Fig. 1. Color denotes MLP confidence: the more saturated, the higher the confidence. Red: noise; green: text

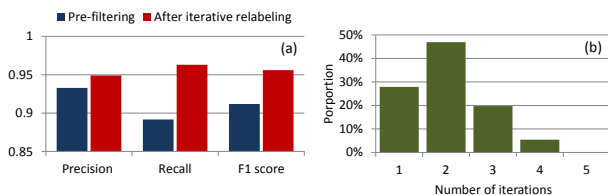


Fig. 8. (a) BB classification rate before and after local iterative relabeling; (b) Number of iterations required for convergence

Deriving a measure of document quality

As shown in the previous subsection, the classifier can label BBs as text or noise with remarkable accuracy, which suggests that it may be used to estimate the overall quality of each document. Low-quality documents tend to produce a large number of spurious BBs, whereas high-quality documents produce mostly text BBs. Thus, the proportion of noise BBs returned by the OCR engine tends to be representative of the document's quality:

$$BB_{noise} = \frac{\# \text{ noise BBs}}{\# \text{ BBs}} \quad (3)$$

We evaluated this quality measure on a large dataset of 6,775 document images from the EEBO collection that had manually-annotated transcriptions. For each document, we computed the similarity s_{JW} between the OCR output and the manual transcription:

$$s_{JW} = 1 - d_{JW} \quad (4)$$

where d_{JW} is the Jaro-Winkler distance (Winkler 1990), a measure of dissimilarity between the two text strings. For the purpose of this work, we used the 'juxta' command-line implementation of the Jaro-Winkler distance available in juxtacommons.org.

Results in Fig. 9(a) show a strong negative correlation (-0.704 ; $p < 0.001$) between the proposed noise measure (BB_{noise}) and the Jaro-Winkler similarity (s_{JW}). Thus, as the proportion of noise BBs in a document increases, so do differences between OCR and manual transcriptions also increase. The significance of this result is that s_{JW} cannot be computed in practice since it requires the manual transcription, whereas BB_{noise} can be computed directly from the output of the OCR engine. As such, it may be used to automatically triage documents of poor quality and focus computational resources on those whose quality is more likely to generate good OCR transcriptions.

Improving OCR transcriptions

In a final step, we tested whether our algorithm could be used to improve the overall OCR performance. For this purpose, we ran the algorithm on the previous dataset (6,775 documents), removed any BBs labeled as noise, and computed s_{JW} between the resulting transcription and the manual transcription. Results are summarized in Fig. 9(b) and Table 3. On 85.4% of the documents the algorithm improved s_{JW} (avg: +6.3%), whereas on 10.6% of the documents it lead to a decrease (avg: -3.0%).

Lastly, we analyzed the impact of local iterative relabeling as a function of document quality; results are shown in Fig. 10. Regardless of document quality (BB_{noise}), local iterative relabeling increases the Jaro-Winkler similarity. These improvements are modest for high-quality documents (i.e., low BB_{noise}), but become

quite significant (up to 0.25) for documents of poor quality, where they are most needed

Discussion

We have presented an approach to assess the quality of OCR using information about the spatial distribution and geometry of word BBs. The approach uses a pre-filtering step to initialize BB labels. From these, the document is segmented into columns by finding troughs in the horizontal distribution of BB coordinates. In a final step, an iterative filtering algorithm is used to incorporate local information from neighboring BBs. When cross-validated on a dataset of 159 historical document images, the algorithm achieves 0.95 precision and 0.96 recall.

The pre-filtering step is designed to minimize false-positive rates since noisy BBs can compromise the subsequent column-segmentation step. As such, the pre-filter tends to miss short text BBs (e.g., short words such as ‘a’, ‘I’, ‘An’) since they violate rule 2. These initial labeling errors are corrected by the iterative relabeling algorithm, which also considers neighborhood information, the relative height of BBs relative to other BBs in the document, and their spatial location in the document. Relabeling generally converges within three iterations, a cost-effective investment considering the improvements in classification performance that it provides.

When evaluated on a collection of documents with manual transcriptions, the proportion of BBs labeled as noise (BB_{noise}) shows a strong correlation with OCR performance, measured as the Jaro-Winkler similarity between OCR and manual transcriptions. As such, BB_{noise} may be used to triage heavily-degraded documents, allowing the OCR engine to focus on documents that have the highest chance of producing accurate transcriptions. Beyond triage, the spatial distribution of noise BBs may be used to provide additional diagnostics for poor-quality documents and direct them to the appropriate process (e.g., rescanning, image denoising). As an example, salt-and-pepper noise tends to generate a large proportion of small BBs, graphics generally result in large and overlapping BBs (see Fig. 7), and marginalia text (see Fig. 6) can be detected by the presence of high-confidence BBs outside the text boundaries. This is particularly important in mass digitization efforts, such as early modern OCR project (eMOP) that motivates this work (Christy, Auvil et al. 2014), where indiscriminate application of image restoration algorithms is prohibitive.

Whenever additional pre-processing (e.g., image restoration) is not viable, our algorithm may still be used to boost OCR accuracy by filtering out noise BBs before the document is submitted for linguistic analysis to correct character recognition errors against historical dictionaries and n-gram models. As illustrated in Table 3 and Fig. 10,

this simple filtering step can lead to significant gains in OCR performance: an average of 6.3% improvement for 85.4% of the documents analyzed. Additional improvements in BB labeling may be obtained by using information from linguistic processing as additional features for the MLP. Denoising then would become an iterative process throughout the post-processing pipeline of improving OCR transcriptions for degraded page images.

Conclusion

Our results indicate that the standard output from an OCR engine (spatial distribution, geometry and confidence of bounding boxes) provides sufficient information to (1) accurately identify text and noise in a document image, (2) estimate the document’s overall quality, and (3) improve OCR transcription performance. This an important result for mass digitization projects, where dedicated image processing becomes prohibitive.

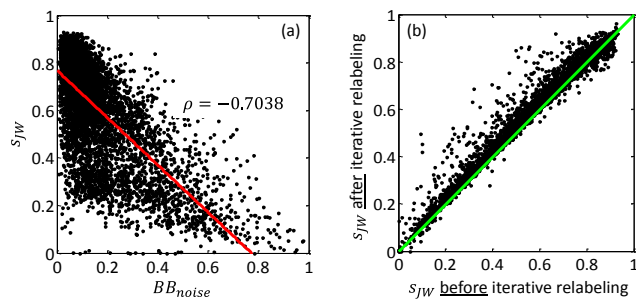


Fig. 9 (a) BB_{noise} -based quality measure (BB_{noise}) vs. the Jaro-Winkler similarity (s_{JW}) for 6,775 documents. (b) s_{JW} before and after iterative relabeling; for most documents (those above the diagonal line) iterative relabeling improved s_{JW}

Table 3 Average change in Jaro-Winkler similarity (Δ) with application of the local iterative relabeling algorithm

	$\Delta > 0$	$\Delta < 0$	$\Delta = 0$
% documents	85.4	10.6	4.0
Avg. change	6.3	3.0	0.0

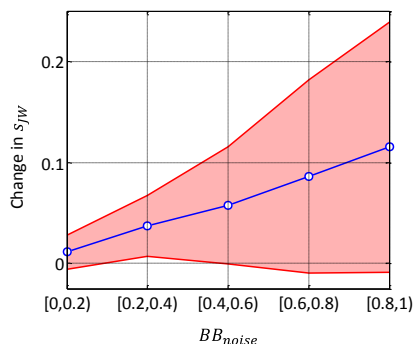


Fig. 10 Average change in Jaro-Winkler similarity as a function of document quality (BB_{noise}).

References

- Alex, B., C. Grover, et al. (2012). Digitised historical text: Does it have to be mediOCRe. Proc. KONVENS 2012 (LThist 2012 workshop)_401-409.
- Ben Salah, A., N. Ragot, et al. (2013). Adaptive detection of missed text areas in OCR outputs: application to the automatic assessment of OCR quality in mass digitization projects. Proc. SPIE 8658, Document Recognition and Retrieval XX. **8658**: 8658161-86581612.
- Christy, M. J., L. Auvil, et al. (2014). Diagnosing Page Image Problems with Post-OCR Triage for eMOP. Proc. Digital Humanities Conference. **in press**.
- Farahmand, A., A. Sarrafzadeh, et al. (2013). Document Image Noises and Removal Methods. Proc. Intl. MultiConference of Engineers and Computer Scientists. **1**: 436-440.
- Liang, J., J. Ha, et al. (1996). Document layout structure extraction using bounding boxes of different entitles. Proc. 3rd IEEE Workshop on Applications of Computer Vision (WACV '96): 278-283.
- Lins, R. D., S. Banergee, et al. (2010). Automatically detecting and classifying noises in document images. Proc. 2010 ACM Symposium on Applied Computing: 33-39.
- Reffle, U. and C. Ringlstetter (2013). "Unsupervised profiling of OCRred historical documents." Pattern Recognition **46**(5): 1346-1357.
- Reynaert, M. (2008). Non-interactive OCR post-correction for giga-scale digitization projects. Computational Linguistics and Intelligent Text Processing, Springer: 617-630.
- Sandhya, N., R. Krishnan, et al. (2012). "A language independent Characterization of Document Image Noise in Historical Scripts." International Journal of Computer Applications **50**(9): 11-18.
- Smith, R. (2007). An Overview of the Tesseract OCR Engine. Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR). **2**: 629-633.
- Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proc. Section on Survey Research Methods (American Statistical Association): 354-359.
- Ye, P. and D. Doermann (2013). Document Image Quality Assessment: A Brief Survey. Proc. 12th Intl. Conf. Document Analysis and Recognition (ICDAR): 723-727.